



**Jorge Augusto  
Fernandes Ferreira**

**Modelação de Sistemas Hidráulicos para Simulação  
com *Hardware-in-the-loop***



**Jorge Augusto  
Fernandes Ferreira**

**Modelação de Sistemas Hidráulicos para Simulação  
com *Hardware-in-the-loop***

dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Doutor em Engenharia Mecânica, realizada sob a orientação científica do Doutor João Pedro Éstima de Oliveira, Professor Associado do Departamento de Engenharia Electrónica e Telecomunicações da Universidade de Aveiro, e do Doutor Fernando Gomes de Almeida, Professor Auxiliar do Departamento de Engenharia Mecânica e Gestão Industrial da Faculdade de Engenharia da Universidade do Porto

**o júri**  
Presidente

DOUTOR SUSHIL KUMAR MENDIRATTA  
Professor Catedrático da Universidade de Aveiro.

DOUTOR CARLOS ALBERTO CARIDADE MONTEIRO COUTO  
Professor Catedrático da Universidade do Minho.

DOUTOR FRANCISCO JORGE TEIXEIRA DE FREITAS  
Professor Associado da Faculdade de Engenharia da Universidade do Porto.

DOUTOR JOÃO PEDRO ESTIMA DE OLIVEIRA  
Professor Associado da Universidade de Aveiro.

DOUTOR FERNANDO GOMES DE ALMEIDA  
Professor Auxiliar da Faculdade de Engenharia da Universidade do Porto.

DOUTOR VÍTOR ANTÓNIO FERREIRA DA COSTA  
Professor Auxiliar da Universidade de Aveiro.

DOUTOR JOSÉ LUÍS COSTA PINTO AZEVEDO  
Professor Auxiliar da Universidade de Aveiro.

## agradecimentos

Quero prestar os meus agradecimentos públicos a pessoas e instituições que contribuíram para que este trabalho fosse possível:

O meu profundo agradecimento aos professores Estima de Oliveira e Fernando Almeida pela orientação e apoio constante durante a realização dos trabalhos relatados nesta dissertação; agradeço-lhes a sua revisão crítica e minuciosa; as correcções e alterações sugeridas em muito valorizaram o seu conteúdo.

Ao professor Rodrigues Quintas pelo apoio na modelação dos sistemas hidráulicos.

Ao professor Vítor Costa pelo incentivo constante e pela sua motivante introdução aos sistemas óleo-hidráulicos.

Ao Eng. Abílio Borges e ao Eng. Paulo Machado pela ajuda na construção da plataforma SCARA usada nas experiências.

Ao professor José Grácio pelo apoio constante e pela disponibilização dos recursos sem os quais este trabalho não teria sido possível.

Aos meus colegas da Modelica Association, em especial ao professor Pieter Mosterman, pelo incentivo à implementação do formalismo dos Statecharts na Linguagem Modelica.

À DYMOsim, na pessoa do seu presidente, professor Hilding Elmqvist, pela disponibilidade da plataforma Dymola, com constante actualização do compilador Modelica.

A todos os colegas do Departamento de Engenharia Mecânica da Universidade de Aveiro, que directa ou indirectamente, contribuíram para a realização deste trabalho.

Gostaria ainda de agradecer à Universidade de Aveiro, instituição onde realizei todo o trabalho.

## resumo

Um ambiente simulado é a via mais rápida, e de menores custos, para avaliar o desempenho de diferentes estratégias para controlo de sistemas hidráulicos. Neste campo emerge uma nova técnica, denominada simulação com *hardware-in-the-loop* (HILS), para teste de controladores reais em ambientes virtuais simulados em tempo real.

A presente dissertação propõe uma metodologia para a organização de bibliotecas de modelos, um conjunto de modelos híbridos semi-empíricos para componentes de sistemas hidráulicos e uma plataforma para a realização de experiências de simulação com *hardware-in-the-loop*.

A metodologia proposta baseia-se na associação de uma nova linguagem de modelação orientada por objectos, chamada Modelica, para a descrição da estrutura dos modelos, com o formalismo gráfico dos Statecharts para a descrição do seu comportamento híbrido. Tendo em vista a concretização da metodologia, foi desenvolvida uma biblioteca de modelos para a implementação do formalismo dos Statecharts em Modelica.

Entre os modelos para componentes hidráulicos propostos está um conjunto de modelos semi-empíricos para a modelação de válvulas proporcionais de elevado desempenho. Os modelos desenvolvidos usam dados do fabricante ou dados experimentais para ajuste dos parâmetros, de forma a reproduzir as características estáticas (ganho de pressão, caudal de fugas e ganho de caudal) e dinâmicas (resposta em frequência) da válvula.

Foi construída uma plataforma para teste do desempenho dos modelos híbridos desenvolvidos em simulações com *hardware-in-the-loop*. A plataforma é composta por um manipulador com actuação hidráulica equipado com um conjunto de transdutores, por hardware de tempo real e por um conjunto de ferramentas de software para aquisição de dados e controlo. De um conjunto de experiências HILS, realizadas com diferentes núcleos de tempo real, foram obtidos resultados muito promissores para os modelos semi-empíricos propostos.

## **abstract**

A simulated environment is the less expensive and fastest way of evaluating the relative merits of different control schemes for a given hydraulic system. A new technique, called hardware-in-the-loop simulation (HILS), is emerging to test hardware controllers in virtual environments simulated in real time.

This thesis proposes a methodology to organize model libraries, a set of semi-empirical hybrid models for hydraulic components and a platform for hardware-in-the-loop simulation experiments.

The methodology holds in the association of a new object oriented modelling language, called Modelica, to describe model structure, with the Statecharts graphical formalism to describe its hybrid behaviour. In order to implement the proposed methodology, a new Statecharts library in Modelica was developed.

Among the proposed hydraulic component models there is a set of new semi-empirical models for high performance proportional valves. The developed models use either data sheet or experimental values to fit the model parameters in order to reproduce both static (pressure gain, leakage flow rate and flow gain) and dynamic (frequency response) valve characteristics.

An experimental platform was setup to test the developed hybrid models performance in hardware-in-the-loop simulation experiments. The platform is composed by a hydraulic actuated manipulator with a set of sensors, by real time hardware and by a package of software tools for data acquisition and control. Very satisfactory results, from a set of HILS experiments performed in different real time kernels were obtained for the semi-empirical models proposed.

À Susy,  
ao João e à Joana

## Lista de símbolos

$A$	área/secção
$a$	aceleração
$C_d$	coeficiente de descarga
$C_p$	calor específico a pressão constante
$C_v$	calor específico a volume constante
$g$	condutância
$f_n$	frequência natural não amortecida
$f_d$	frequência natural amortecida
$F, f$	forças
$f_{co}$	força de atrito de Coulomb
$f_s$	força de atrito de Stribeck
$f_{bw}$	atrito estático (força de <i>break-away</i> )
$k_v$	coeficiente de atrito viscoso
$\bar{K}_{p0}$	ganho de pressão relativa em $\bar{x}_s = 0$
$\bar{K}_{q0}$	ganho de caudal para $\bar{x}_s = 0$
$L_v, L_a$	limites de velocidade e aceleração
$J$	momento de inércia
$M$	massa
$P$	pressão relativa
$\bar{P}$	pressão relativa à pressão de trabalho
$\Delta P$	diferença de pressão
$P_L$	diferença de pressão na carga
$\bar{P}_L$	diferença de pressão relativa na carga ( $\bar{P}_L = P_L/(\text{pressão de trabalho})$ )
$P_n$	diferença de pressão nominal
$Q, q$	caudais volumétricos
$Q_L$	caudal na carga
$Q_n$	caudal nominal
$t$	tempo
$T$	temperatura
$\bar{u}$	senal de entrada normalizado; $\bar{u} \hat{=} [-1, 1]$
$V$	volume
$V_L$	volume em linhas de transmissão



$v$	velocidade
$x$	posição
$\bar{x}_s$	posição normalizada; $\bar{x}_s \in [-1, 1]$
$q$	ângulo
$\xi$	coeficiente de amortecimento
$\beta$	módulo de compressibilidade
$\beta_e$	módulo de compressibilidade efectivo
$\rho$	massa específica volúmica {densidade}
$m$	viscosidade dinâmica
$h$	viscosidade cinemática
$\omega$	velocidade angular
$\omega_n$	frequência angular natural
$\Delta t$	atraso temporal

## Índices

$0$	valor inicial
$in$	entrada
$ij$	de $i$ para $j$
$k$	valor discreto
$lam$	regime laminar
$lk$	fugas
$max$	valor máximo
$min$	valor mínimo
$out$	saída
$ref$	valor de referência
$s$	fonte (em caudais e pressões)
$ss$	valor em regime estacionário
$T$	regime turbulento
$t$	tanque (em caudais e pressões)

## Abreviaturas

BG	Bond Graphs
BDF	Backward Difference Formula
CSSL	Continuous System Simulation Language
DAE	Equações Diferenciais e Algébricas
ECU	Unidade de controlo electrónica
FSA	Automata de Estados Finitos
FSM	Máquina de Estados Finita
HA	Automata Híbrido
HILS	Simulação com Hardware-In-the-Loop
ODE	Equações Diferenciais Ordinárias
PDE	Equações Diferenciais Parciais
PI	Controlo Proporcional + Integral
PID	Controlo Proporcional + Integral + Derivativo
PN	Redes de Petri
STR	Simulação em Tempo Real
VI	Índice de Viscosidade

Os termos seguintes, apresentados ao longo desta dissertação, são marcas registradas.

ACSL	Rhapsody
ADAMS	RealSim
Amesim	ROOM
ASCEND	RT-LAB
BetterState	SCANCON
Bosch	Shift
C++	Siman
Chi	Simple++
CSSL	Simula
dSPACE	Simulink
Dymola	Smile
Easy5	Stateflow
Eiffel	Statemate
FGP Instrumentation	SystemBuild
gProms	Tellus68
HBM	TEXTRON
Humusoft	UML
Mathwoks	Verilog
Matlab	VHDL
Modelica	VHDL-MAS
MTS	Vickers
National Instruments	VisSim
Objectcharts	Visual C/C++
ObjectMath	Watcom C/C++
OLAER	Webster Instruments
Omola	Windows

## ÍNDICE

<b>1</b>	<b>INTRODUÇÃO</b>	<b>I</b>
<b>2</b>	<b>MODELAÇÃO E SIMULAÇÃO DE SISTEMAS FÍSICOS</b>	<b>10</b>
2.1	Terminologia e conceitos	10
2.2	Modelos, modelação e simulação	12
2.2.1	Modelos matemáticos: ODEs e DAEs	14
2.2.2	Solução numérica	16
2.3	Sistemas híbridos	19
2.3.1	Modelos para sistemas híbridos	20
2.3.2	Simulação de modelos híbridos	22
2.3.3	Exemplo de um sistema híbrido: servomecanismo hidráulico	24
2.4	Linguagens de modelação e simulação	29
2.5	Formalismos para descrição de comportamento	34
2.6	A linguagem Modelica	38
<b>3</b>	<b>METODOLOGIA DE MODELAÇÃO: STATECHARTS E MODELICA</b>	<b>54</b>
3.1	Statecharts	54
3.1.1	Considerações informais sobre a sintaxe e a semântica dos Statecharts	54
3.1.2	Conflito em transições e não determinismo	57
3.1.3	O conceito de história	58
3.1.4	Representação da estrutura de um statechart	59
3.1.5	Ações e actividades para modelação de comportamentos híbridos	60
3.1.6	A semântica de execução	61
3.1.7	Especificação sintáctica formal	63
3.1.8	Herança de comportamento nos Statecharts	66

<b>3.2</b>	<b>Implementação do formalismo dos Statecharts em Modelica</b>	<b>66</b>
3.2.1	Utilização da hipótese síncrona	67
3.2.2	Implementação da hierarquia dos statecharts	68
3.2.3	Dois níveis de implementação	69
3.2.4	Mecanismos para a implementação da história	71
3.2.5	Bibliotecas de modelos	71
3.2.6	Utilização da biblioteca de modelos	76
<b>3.3</b>	<b>Metodologia de modelação proposta</b>	<b>78</b>
3.3.1	Metodologia para a modelação	78
3.3.2	Metodologia para organização de bibliotecas de modelos	83
<b>4</b>	<b>MODELAÇÃO DE SISTEMAS HIDRÁULICOS</b>	<b>88</b>
<b>4.1</b>	<b>Fluidos hidráulicos e suas características</b>	<b>88</b>
4.1.1	Massa específica volúmica	88
4.1.2	Módulo de compressibilidade	89
4.1.3	Viscosidade	96
<b>4.2</b>	<b>Modelos de actuadores</b>	<b>98</b>
<b>4.3</b>	<b>Modelação do atrito</b>	<b>105</b>
4.3.1	Atrito em regime de velocidade estacionária (modelos estáticos)	106
4.3.2	Modelos dinâmicos	116
<b>5</b>	<b>MODELOS PARA VÁLVULAS PROPORCIONAIS</b>	<b>132</b>
<b>5.1</b>	<b>Modelos estáticos</b>	<b>134</b>
5.1.1	Características estáticas da válvula	137
5.1.2	Modelos com escoamento misto	141
5.1.3	Modelos com escoamento turbulento	151
<b>5.2</b>	<b>Modelos dinâmicos</b>	<b>179</b>
5.2.1	Ajuste do ganho de amplitude	182
5.2.2	Modelos e simulações	183
5.2.3	Ajuste de fase	189

<b>6</b>	<b>EXPERIÊNCIAS DE SIMULAÇÃO COM <i>HARDWARE-IN-THE-LOOP</i></b>	<b>192</b>
6.1	Plataforma experimental	192
6.2	Identificação de parâmetros de alguns modelos	199
6.2.1	Medição indirecta do módulo de compressibilidade	199
6.2.1.1	Modelo linear do conjunto válvula + cilindro	201
6.2.1.2	Experiências para determinação da frequência natural amortecida	206
6.2.1.3	Estimação do módulo de compressibilidade efectivo	207
6.2.1.4	Parametrização dos modelos para o módulo de compressibilidade efectivo	210
6.2.2	Medição da condutância de fugas internas do cilindro	211
6.2.3	Parametrização do modelo de atrito de LuGre para o cilindro	211
6.2.3.1	Identificação dos parâmetros estáticos	212
6.2.3.2	Identificação dos parâmetros dinâmicos	213
6.3	<i>Hardware-in-the-loop</i>	217
6.3.1	Técnicas e métodos numéricos na STR	218
6.3.2	Aplicações e plataformas para HILS	221
6.3.3	Exemplos de aplicações de HILS	222
6.3.4	Plataformas HILS implementadas	222
6.3.5	Experiências de HILS	225
6.3.5.1	Experiências HILS na plataforma dSPACE	225
6.3.5.2	Experiências HILS na plataforma RTWT	231
6.3.5.3	Experiências HILS na plataforma xPC	236
<b>7</b>	<b>CONCLUSÕES FINAIS</b>	<b>244</b>
<b>8</b>	<b>REFERÊNCIAS</b>	<b>249</b>

# 1 INTRODUÇÃO

Os sistemas hidráulicos são, de há muito, usados em larga escala nas indústrias de manufactura e em maquinaria pesada. Os componentes hidráulicos sofreram uma grande evolução nos últimos anos, desde os componentes hidro-mecânicos até aos sofisticados sistemas electro-hidráulicos controlados por microprocessador. O controlo deste tipo de sistemas era inicialmente realizado em malha aberta, com a percepção humana a ser o único sinal de realimentação. Com a crescente necessidade de precisão e automação dos sistemas hidráulicos, técnicas variadas foram usadas para introduzir realimentação neste tipo de sistemas. Inicialmente foram usadas as realimentações mecânica ou hidráulica para resolver problemas específicos; por exemplo, servoválvulas hidráulicas foram empregues em sistemas de posicionamento. Com a evolução tecnológica o uso da electrónica e dos microprocessadores contribuiu para aumentar o desempenho e acrescentar novas funcionalidades aos sistemas hidráulicos tradicionais.

Muito embora tenha havido uma grande evolução no hardware dos componentes hidráulicos, o projecto de sistemas hidráulicos é, ainda hoje, muito baseado na tradição e na experiência. Este facto, aliado à crescente complexidade dos sistemas hidráulicos modernos, pode levar a comportamentos inesperados e a erros de projecto. Em muitos casos, o desempenho dos sistemas hidráulicos é avaliado com programas de simulação em computador, normalmente escritos em FORTRAN, ou então recorrendo a protótipos físicos [Ellman95].

Os sistemas hidráulicos são, na sua generalidade, complexos e não lineares devido a variados factores de que se destacam a natureza laminar ou turbulenta dos escoamentos, a geometria dos componentes, o atrito introduzido pelos vedantes, as forças induzidas pelos escoamentos ou as propriedades inerentes aos óleos que variam com a temperatura e com a pressão. Devido a estas dinâmicas complexas e às não linearidades deste tipo de sistemas, os algoritmos de controlo normalmente aplicados a sistemas lineares, como os tradicionais algoritmos PID, podem apresentar desempenhos pouco satisfatórios em aplicações hidráulicas mais exigentes. Nos últimos anos houve uma grande evolução no controlo de sistemas e dispositivos não-lineares, que induziu o estudo e utilização de novas técnicas de controlo em sistemas hidráulicos. Exemplos dessas

estratégias são o controlo adaptativo [Bobrow96], o controlo por lógica difusa [Koskinen92], ou o controlo por redes neuronais [Newton95]; Edge apresenta uma análise comparativa de diferentes estratégias de controlo para sistemas hidráulicos [Edge97]. Este interesse é mais patente hoje em dia no uso de estratégias para aumentar a precisão e o desempenho dinâmico de maquinaria industrial sofisticada, nomeadamente em máquinas-ferramenta de alta velocidade.

A experimentação de novos algoritmos de controlo directamente no hardware hidráulico não é, contudo, um método apropriado por diversas razões, em particular o custo e as dimensões do hardware e as suas condições de funcionamento, para além das questões de segurança. No entanto, é importante avaliar os méritos e deméritos das diferentes estratégias de controlo para uma dada aplicação, sendo a simulação em computador um dos principais meios de avaliação. De facto, um ambiente simulado é a forma mais rápida e de mais baixo custo para testar algoritmos de controlo. Para muitas aplicações é praticamente impossível reproduzir em laboratório os sistemas hidráulicos em conjunto com adequadas condições de operação. Os trabalhos realizados por Gonthier [Gonthier98], DiMaio [DiMaio98] ou Schothorst [Schothorst97] são exemplos de reproduções do hardware de maquinaria hidráulica e das suas condições de operação através de simulações e visualização em 3D do seu comportamento dinâmico. Estes trabalhos caracterizam-se pela utilização de filosofias e metodologias de modelação próprias, complicando o intercâmbio e a actualização dos modelos noutros ambientes.

Uma uniformização das metodologias e dos ambientes de modelação permite uma mais fácil modelação de sistemas complexos e a re-utilização de modelos, pelo que a representação de modelos de uma forma sistemática e flexível tem vindo a ser estudada nos últimos anos. As soluções adoptadas recorrem a linguagens de modelação de índole específica e de índole geral [Otter95]. As metodologias e as filosofias usadas nas linguagens são diversas, podendo, contudo, destacar-se duas categorias: as técnicas de programação tradicionais e as metodologias orientadas por objectos. Hoje em dia, mais atenção é concedida às linguagens orientadas por objectos pois, com estas, é mais fácil a re-utilização, a expansão ou a adaptação de modelos já desenvolvidos. A confirmar esta tendência está o trabalho realizado por Beater [Beater98], que refere que a parte que consome mais tempo na modelação de sistemas pode ser acelerada consideravelmente através do uso das linguagens de modelação orientadas por objectos e bibliotecas de componentes.

Os componentes hidráulicos são, muitas vezes, parte de sistemas complexos que podem também incorporar sistemas mecânicos, eléctricos, electrónicos e, ainda, software. De facto, o maior obstáculo na definição de técnicas de modelação e de simulação é este cenário complexo e a multi-disciplinaridade deste tipo de sistemas. A integração de modelos de diferentes domínios é também uma tarefa não trivial e que consome bastante tempo pois, muito embora existam bibliotecas de



modelos bastante desenvolvidas para os diversos domínios, elas são geralmente descritas em linguagens diferentes e, quase invariavelmente, incompatíveis. Devido à multi-disciplinaridade e à complexidade dos sistemas hidráulicos modernos, a simulação de todos os componentes de forma contínua torna-se bastante pesada e morosa, incompatível com as necessidades da engenharia moderna. Por isso, considera-se que alguns componentes hidráulicos contínuos, por exemplo válvulas *on-off*, podem ser modeladas através de modelos discretos, na maioria das aplicações. Este tipo de sistemas, cujo comportamento dinâmico é híbrido, tem também vindo a ser alvo de investigação, da qual têm resultado é uma série de linguagens de modelação que permitem a descrição desses sistemas. Diversos formalismos gráficos também têm vindo a ser utilizados para a descrição do comportamento de sistemas híbridos.

A modelação e a simulação em tempo real de sistemas complexos é ainda, segundo Burrows [Burrows98], uma área a explorar. De facto, e de acordo com Lennevi [Lennevi95], com o aumento do poder de cálculo sistemas cada vez mais complexos podem ser simulados em tempo real com custos menores. A simulação com *hardware in-the-loop* tem vindo a ser usada com sucesso na indústria aeroespacial e está agora a emergir como uma técnica de teste de unidades de controlo. Esta técnica tem sido aplicada na resolução de alguns problemas específicos, mas a tendência é usá-la como uma plataforma de teste do comportamento em tempo real de componentes de hardware.

No seu essencial, este trabalho propõe uma metodologia para a modelação e organização de bibliotecas de modelos de componentes hidráulicos tendo como finalidade a realização de experiências com *hardware-in-the-loop*. É comumente aceite que modelos mais complexos exigem desempenhos computacionais e tempos de simulação maiores, sendo apropriada a simulação em tempo não real. O esforço incide então na procura de modelos de baixa complexidade, que poderão ser semi-empíricos, mas que traduzam de forma eficaz os comportamentos dos componentes, muito especialmente os mais importantes para o teste de controladores. É então proposto um conjunto de modelos de diferentes complexidades, para os componentes de um sistema hidráulico, de modo a ajustar a complexidade do modelo global ao hardware de simulação em tempo real disponível e, assim, possibilitar a realização de experiências com *hardware-in-the-loop* permitindo, dessa forma, o teste de novos controladores e/ou algoritmos sem haver a necessidade de montagem de protótipos de hardware.

A presente dissertação está organizada em 7 capítulos constituindo esta introdução o primeiro capítulo.

O Capítulo 2 começa por abordar a terminologia da modelação e simulação de sistemas físicos e de engenharia de uma forma genérica. Introduce a problemática da modelação de sistemas com

comportamentos contínuos e discretos (sistemas híbridos), bem como o esforço dispendido nos últimos anos no desenvolvimento de teorias, aplicáveis a este tipo de sistemas, traduzidas em propostas de linguagens e formalismos de modelação. Introduce os conceitos de uma nova linguagem de modelação orientada por objectos, chamada Modelica, adequada à modelação de sistemas de vários domínios físicos e que possam envolver comportamentos híbridos.

O Capítulo 3 apresenta a metodologia proposta para a modelação e organização de bibliotecas de modelos de componentes de sistemas hidráulicos. A metodologia assenta na associação de uma linguagem de modelação orientada por objectos (Modelica), para a descrição da estrutura dos modelos, com um formalismo gráfico (Statecharts), adequado à descrição do comportamento dinâmico de sistemas híbridos. É apresentado o desenvolvimento de uma biblioteca de modelos para implementação do formalismo dos Statecharts em Modelica, tendo em vista a concretização da metodologia com uma única linguagem de modelação.

O Capítulo 4 descreve um conjunto de modelos de caracterização do fluido hidráulico, muito especialmente a dependência do módulo de compressibilidade efectivo com a pressão. Aborda também as técnicas de modelação de actuadores hidráulicos utilizando modelos de parâmetros concentrados e recorre ao formalismo dos Statecharts para a representação de limites de curso, indispensáveis à simulação em tempo real. Implementa um conjunto de modelos estáticos e dinâmicos de atrito recorrendo, sempre que necessário, ao formalismo dos Statecharts como uma ferramenta útil na especificação dos comportamentos híbridos dos modelos.

No Capítulo 5 são propostos vários modelos semi-empíricos para modelação estática e dinâmica de válvulas proporcionais. A parametrização destes modelos assenta essencialmente em medidas normalmente fornecidas nos catálogos dos fabricantes, nomeadamente medidas de caudal, de pressão e respostas em frequência.®

No Capítulo 6 são apresentadas algumas experiências realizadas com *hardware-in-the-loop* em diferentes plataformas de baixo custo. Inicialmente é proposto um conjunto de técnicas para parametrização dum conjunto válvula e actuator linear, que permite aferir a qualidade e o desempenho dos modelos propostos, por comparação com um sistema real equivalente. De seguida são abordadas as técnicas, métodos numéricos adequados e um conjunto de plataformas de simulação em tempo real de baixo custo. É apresentada a plataforma experimental desenvolvida, incluindo o seu software de suporte, para a parametrização de modelos e a comparação com resultados de simulação. São relatadas experiências de *hardware-in-the-loop*, com complexidades diferentes, em plataformas baseadas em processadores digitais de sinal e em plataformas baseadas em computador pessoal.

No último capítulo são apresentadas conclusões e algumas perspectivas de desenvolvimento futuro do trabalho realizado.

## 2 MODELAÇÃO E SIMULAÇÃO DE SISTEMAS FÍSICOS

### 2.1 Terminologia e conceitos

Encontram-se na literatura definições variadas para caracterizar um **sistema** [Ashby56], [Zeigler76], [Cellier91], [Mosterman97]. Pode-se definir um sistema como uma potencial fonte de dados resultantes do agrupamento de um conjunto de fenómenos que podem ser observáveis e controláveis. O conjunto de fenómenos, o seu controlo e as observações feitas no sistema determinam a sua fronteira. Tudo o que não pertence ao sistema é chamado o seu ambiente e as interacções sistema/ambiente definem o contexto do sistema. As interacções de um sistema com o ambiente podem ser organizadas em duas categorias: as variáveis geradas pelo ambiente, e que influenciam o comportamento do sistema, designam-se por entradas do sistema; as variáveis que são determinadas pelo sistema, e que influenciam o comportamento do seu ambiente, são chamadas as saídas do sistema (Figura 2.1). Uma experiência pode ser vista como um processo de extracção de dados de um sistema, através da excitação das suas entradas e observação das suas saídas.

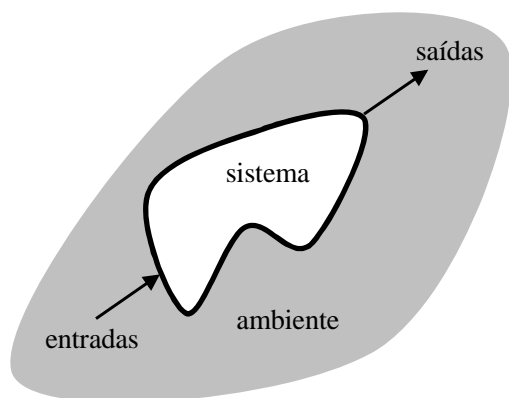


Figura 2.1 Conceito de sistema

A **teoria dos sistemas** [Luenberger79], [Katok95] é uma matéria interdisciplinar com incidência sobre a descrição do comportamento dinâmico de sistemas, normalmente realizada através da utilização das leis da física. Isaac Newton foi o primeiro a registar a teoria formal no domínio da mecânica, tendo sido desenvolvidas posteriormente as teorias para outros domínios, como, entre

outros, a mecânica dos fluidos, a electricidade e o magnetismo. A teoria termodinâmica dos sistemas físicos teve a sua génese em observações que indicavam que diferentes domínios físicos podem interagir entre eles através de intercâmbio energético. O desenvolvimento dessas teorias veio provar que diferentes sistemas reais podem obedecer às mesmas leis físicas e podem apresentar comportamentos idênticos. Isto levou a que alguns sistemas, muito embora possam ser fisicamente distintos, viessem a ser representados em termos do equivalente eléctrico ou usando representações independentes dos sistemas como, por exemplo, as redes de ligações energéticas (Bond Graphs) [Paynter61]. Podem então ser usados conjuntos de componentes interligados, tais como fontes de energia, dissipadores, elementos armazenadores de energia ou elementos conversores de formas de energia para a descrição de sistemas em domínios diferentes. Mais recentemente foram estabelecidas as teorias do controlo e da informação. A teoria dos sistemas permite unificar e generalizar as teorias dos vários domínios numa descrição matemática comum, normalmente sob a forma de equações diferenciais. Esta descrição matemática comum permite a descrição do comportamento de sistemas que envolvem componentes de diferentes domínios físicos, situação recorrente nos sistemas de engenharia.

Por comportamento do sistema entende-se a forma como o sistema se manifesta na sua fronteira, isto é, como reage às entradas gerando as saídas. O interior de um sistema pode ser descrito pelo seu estado e pela sua dinâmica; o estado representa as condições em que o sistema se encontra num determinado instante enquanto a dinâmica descreve a evolução do estado ao longo do tempo. Nos sistemas físicos, o estado é normalmente descrito por um conjunto de quantidades contínuas, conhecidas como variáveis de estado, denominado por vector de estado,  $x$ . A dinâmica do sistema pode ser representada por um conjunto de funções de transição de estado,  $f$ , as quais podem depender do conjunto de entradas,  $u$ , e do próprio estado. A derivada do vector de estado,  $\dot{x}$ , é então usada para descrever o comportamento do sistema:  $\dot{x} = f(x, u)$ . Todas as outras variáveis,  $s$ , estão relacionadas de forma algébrica com o estado e com as entradas, por um conjunto de funções matemáticas,  $h$ , na forma:  $s = h(x, u)$ .

O vector de estado representa o núcleo da descrição do sistema captando completamente o historial do seu comportamento, isto é, representando a memória do passado do sistema, de tal forma que deve ser possível, em qualquer instante, calcular o futuro conhecendo somente o estado e as entradas futuras [Zeigler00]. Então, as variáveis de estado representam a parte que deve ser inicializada no início de cada experiência de tal forma que, quando esta é interrompida, as variáveis de estado são suficientes para continuar a experiência e conseguir os mesmos resultados que se obteriam se a experiência não tivesse sido interrompida.

## 2.2 Modelos, modelação e simulação

A descrição do comportamento dos sistemas físicos pode ser sistematizada numa forma mais abstracta através de um **modelo**; um modelo formaliza então o conhecimento sobre um dado sistema. Citando Rothenberg [Rothenberg89] “um modelo representa a realidade para um determinado propósito; o modelo é uma abstracção da realidade no sentido em que não pode representar todos os aspectos da realidade. Isto permite lidar com os sistemas de uma maneira simplificada, evitando a complexidade, o perigo, e a irreversibilidade da realidade”. Pode-se então definir um modelo como sendo uma descrição de como um sistema se comporta. Em vez de se interagir com um sistema real, pode-se criar um modelo que lhe corresponda em certos aspectos. Praehofer [Praehofer91] refere que um modelo representa de uma forma válida um sistema real de acordo com objectivos particulares; de facto, um modelo pode ser válido para “responder” a uma questão particular do sistema real e pode falhar noutra tipo de questões.

Um modelo matemático de um sistema físico pode, por si próprio, fornecer informação sobre o sistema. Se o modelo puder ser simulado mais rapidamente que a evolução natural do sistema, então este pode ser usado para prever o comportamento futuro do sistema. O projecto de novos dispositivos de certa complexidade requer uma modelação prévia para assegurar que o dispositivo fornece o comportamento desejado. A determinação de erros num sistema que não funciona pode ser simplificada pela existência de um modelo do sistema. Um modelo pode também ser útil no projecto de plataformas de teste para o sistema real.

Ao processo de organização de conhecimento de um determinado sistema, isto é, ao processo de construção de modelos chama-se **modelação**.

A modelação de sistemas físicos deve incluir as características salientes que determinam o comportamento dinâmico dos sistemas. Um factor muito importante é que os sistemas físicos são dependentes do tempo, podendo o seu comportamento, na maioria dos casos, ser definido por equações diferenciais e algébricas. A construção de um modelo significa sempre idealização, abstracção e simplificação. A questão está em saber até que ponto o modelo, sendo uma representação idealizada e simplificada, é ou não uma representação válida do sistema real; a qualidade de um modelo pode então ser averiguada pela forma como o seu comportamento se aproxima da realidade em causa. A tarefa de criação de modelos é muitas vezes elaborada requerendo conhecimentos profundos na área em causa. Essa tarefa também é útil na organização e estruturação do conhecimento de modo a obter uma compreensão mais profunda do sistema.

À realização de uma experiência num modelo chama-se uma **simulação**. A simulação consiste numa construção de um outro sistema real que pode ser visto como uma implementação do modelo

e oferece a possibilidade de realização de experiências com ele. Este segundo sistema real, que se comporta de forma equivalente ao primeiro é, muitas vezes, chamado modelo de simulação. Segundo Schmidt [Schmidt87], os modelos de simulação podem ser incluídos em três categorias: modelos de simulação físicos, modelos de simulação gráficos e modelos de simulação em computador. Os modelos de simulação físicos correspondem a protótipos físicos, normalmente construídos a escalas diferentes, que permitem experimentações mais fáceis. Os modelos gráficos correspondem a representações gráficas de sistemas reais. Os modelos de simulação por computador recorrem a computadores analógicos ou digitais como um meio de reproduzir o comportamento do sistema real; hoje em dia, os computadores digitais são, na maioria dos casos, o meio usado para a implementação dos modelos de simulação por computador. De facto, os métodos mais avançados de projecto em engenharia baseiam-se na modelação e simulação baseadas em computador de modo a evitar os elevados custos no desenvolvimento de protótipos físicos [Norris95].

A modelação e simulação têm-se tornado cruciais em todas as disciplinas de ciência e engenharia. Estas “ferramentas” têm sido usadas na análise de sistemas físicos pois permitem uma melhor compreensão dos fenómenos, sendo também muito importantes no desenho de novos sistemas de engenharia, pois permitem prever o comportamento do sistema previamente à sua construção. De facto, e de acordo com Cellier [Cellier91], a modelação e a simulação são as únicas técnicas disponíveis que permitem uma análise arbitrária e precisa de sistemas não lineares e com condições de experimentação variáveis. A modelação e a simulação são também fundamentais para o desenvolvimento de sistemas de controlo, já que, para processos complexos e não lineares, não existem métodos analíticos adequados. Para Elmqvist [Elmqvist93] a simulação pode ser, na maioria dos casos, o único recurso disponível para construir o sistema e experimentar os controladores.

A Figura 2.2 apresenta uma classificação dos modelos de simulação de acordo com Praehofer [Praehofer91]. Praehofer refere que a temporização do modelo e a trajectória das variáveis podem ser usadas para a classificação dos modelos de simulação. De acordo com o critério tempo, podem-se classificar os modelos em tempo contínuo, quando o tempo é representado através de números reais, e em tempo discreto, quando o tempo é representado por inteiros ou um conjunto limitado de reais. De acordo com a mudança de trajectória das variáveis, os modelos são classificados em variação contínua, quando as variações do estado ocorrem continuamente ao longo do tempo, ou então são classificados em variação discreta, quando ocorrem em transições discretas. As combinações de categorias, correspondentes aos dois critérios, definem diferentes abordagens de modelação.

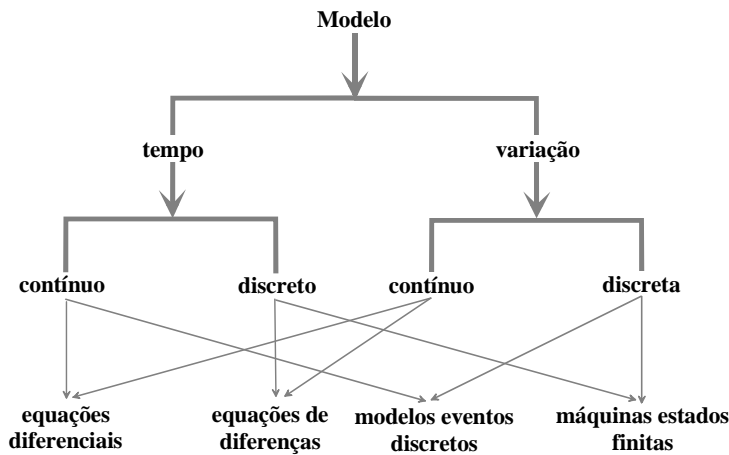


Figura 2.2 Classificação dos modelos de simulação

### 2.2.1 Modelos matemáticos: ODEs e DAEs

Os modelos matemáticos de sistemas físicos resultam da aplicação das leis da física que descrevem, com alguma aproximação, o comportamento desses sistemas. Uma descrição matemática precisa é de grande utilidade quer em termos descritivos quer em termos preditivos. Para muitas aplicações, como o caso do presente trabalho, são usados modelos de parâmetros concentrados, onde os atributos físicos dos sistemas, como a inércia, rigidez ou amortecimento, são assumidamente discretos no espaço ao contrário dos modelos de parâmetros distribuídos onde estes atributos são contínuos. A descrição matemática mais familiar resultante dos modelos de parâmetros concentrados baseia-se num sistema de equações diferenciais ordinárias (ODEs) [Braun83], [Dreyer93].

Um sistema de equações diferenciais ordinárias tem a forma geral:

$$\dot{x} = f(x, t) \tag{2.1}$$

onde  $t \in \mathbb{R}$ ,  $x \in \mathbb{R}^n$  é um vector de variáveis,  $f$  é uma função com domínio em  $\mathbb{R}^{n+1}$  e contradomínio em  $\mathbb{R}^n$  ( $f : \mathbb{R}^{n+1} \rightarrow \mathbb{R}^n$ ), e  $\dot{x} = \frac{dx}{dt}$  é a derivada de  $x$  em ordem a  $t$ . Isto é:

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \vdots \\ \dot{x}_n \end{pmatrix} = \begin{pmatrix} f_1(x_1, x_2, \dots, x_n, t) \\ f_2(x_1, x_2, \dots, x_n, t) \\ \vdots \\ f_n(x_1, x_2, \dots, x_n, t) \end{pmatrix} \tag{2.2}$$

A equação (2.1) define um sistema de equações diferenciais acopladas onde se conhece a função  $f$  e se pretende determinar a variável desconhecida  $x$ .



A equação diferencial ordinária (2.1) não determina, por si só, uma solução pois a equação somente especifica o declive  $\dot{x}(t)$  da solução em cada ponto e não o seu valor actual  $x(t)$ . Em geral, existe um número infinito de soluções, sendo necessário especificar um valor para a solução,  $x_0$ , num determinado instante,  $t_0$ , de tal forma que:  $x(t_0) = x_0$ .

Este requisito adicional determina uma solução única para o sistema de ODE, desde que  $f$  tenha derivada contínua. A variável independente  $t$  normalmente representa o tempo, definindo-se  $t_0$  como o instante inicial e  $x_0$  como o valor inicial. O ODEs descreve então a evolução dinâmica do sistema no tempo, desde o seu estado inicial  $x_0$  no instante  $t_0$ .

Uma descrição mais geral para os sistemas dinâmicos assenta nos sistemas de equações diferenciais e algébricas (DAEs), que, basicamente, são um conjunto de equações diferenciais ordinárias com constrangimentos algébricos. Na sua forma mais geral, um DAEs é descrito por um conjunto de equações

$$f(\dot{x}, x, t) = 0 \tag{2.3}$$

onde  $t \in \mathbb{R}$  é a variável independente,  $x \in \mathbb{R}^n$  é um vector de variáveis,  $f: \mathbb{R}^{n+1} \rightarrow \mathbb{R}^m$  e  $\dot{x} = dx/dt$  é a derivada de  $x$  em ordem a  $t$ .

A equação (2.3) representa um DAEs na sua forma implícita. Se os constrangimentos algébricos do DAEs forem explícitos, então tem-se um DAEs semi-explícito

$$\begin{aligned} f(\dot{x}, x, y, t) &= 0 \\ g(x, y, t) &= 0, \end{aligned} \tag{2.4}$$

onde o Jacobiano de  $f$  em relação a  $\dot{x}$  ( $\partial f / \partial \dot{x}$ ) é não singular.

As variáveis são então divididas em duas categorias. As variáveis  $x$  são chamadas variáveis diferenciais e as variáveis  $y$  são chamadas variáveis algébricas.

Se a equação (2.3) puder ser re-escrita numa forma explícita,  $\dot{x} = F(x, t)$ , então neste caso (2.3) é um ODEs representado na sua forma implícita. Contudo, a transformação de um DAEs num ODEs nem sempre é possível ou desejável, pois requer alguma manipulação ou simplificação das equações, o que pode reduzir a expressividade do modelo. Então, em algumas aplicações, é desejável a resolução do DAEs directamente.

Os DAEs são caracterizados por uma propriedade importante, denominada índice do DAEs. Na literatura podem-se encontrar várias definições para o índice de um DAEs. Gear, em [Gear 88], define o índice de um DAEs como o número mínimo de vezes que todas as partes de  $f(\dot{x}, x, t) = 0$  têm que ser diferenciadas de modo a determinar  $\dot{x}$  em função de  $x$  e  $t$ . Um ODEs apresentado no

formato de espaço de estados,  $\dot{x} = F(x, t)$ , tem um índice zero. Um DAEs apresentado no formato (2.4) tem índice um se o Jacobiano de  $f$  em relação a  $x$  ( $\partial f / \partial x$ ) for não singular. Um DAEs no formato geral  $f(x, t) = 0$  pode ter um índice mais elevado; índices elevados indicam que o modelo tem relações algébricas entre variáveis diferenciáveis. Porém, o índice não é uma propriedade do sistema modelado mas sim da forma como este é modelado, sendo possível reduzir o índice através de manipulações algébricas [Matsson93], [Fábian00].

## 2.2.2 Solução numérica

A solução numérica dos ODEs e DAEs encontra-se bem documentada em [Ascher98], [Teukolsky92] ou [Byrne92] e é essencial nos simuladores implementados em computadores digitais. Existem variadas formas de calcular soluções aproximadas para sistemas de equações diferenciais ordinárias apresentados como em (2.1). Os métodos assentam na substituição das equações diferenciais por equações às diferenças. O método mais simples é baseado na aproximação da derivada por equações às diferenças de 1ª ordem e é conhecido como o método de Euler:

$$x_{k+1} = x_k + \Delta t_k \times f(x_k, t_k) \quad (2.5)$$

O método de Euler despreza os termos da série de Taylor a partir de segunda ordem para obter o valor aproximado para a variável de estado num instante  $t_{k+1} = t_k + \Delta t_k$ . O método de Euler é conhecido como um método de passo simples, pois depende da informação num só ponto no tempo para avançar para o próximo ponto. Técnicas mais eficientes são os métodos de Runge-Kutta ou os métodos de passo variável.

A precisão é um aspecto bastante importante na selecção de um método adequado. O método de Euler é um método com implementação extremamente simples mas tem baixa precisão e requer normalmente passos pequenos. Em alguns casos o passo  $\Delta t_k$  tem de ser tão pequeno que os erros de arredondamento tornam-se importantes. Métodos de ordem mais elevada produzem resultados mais precisos, aproximando a solução em cada passo por um polinómio de maior grau. Contudo, isto requer, normalmente, o uso de derivadas de ordem mais elevada.

Além da precisão, outro aspecto importante a considerar na integração de um ODEs é a estabilidade. Pode-se definir estabilidade de um método como a propriedade do método limitar os erros resultantes do processo de integração ao longo dos passos de integração. Um método instável origina crescimento exponencial dos erros de integração, podendo provocar *overflow* aritmético ao fim de alguns passos. O conceito de estabilidade de um ODEs  $\dot{x} = f(x, t)$  pode ser melhor

quantificado por análise dos valores próprios da matriz Jacobiana  $\{J\}_{ij} = \partial f_i / \partial x_j$ . A instabilidade do sistema ocorre quando as partes reais dos valores próprios se tornam positivas. Como os elementos da matriz  $J$  são funções de  $t$  e  $x$ , os seus valores próprios podem variar com o tempo e, assim, a estabilidade pode alterar-se ao longo do processo de integração.

Informação complementar sobre implementação, vantagens e desvantagens, estabilidade, controlo de erro, etc, dos diferentes métodos pode ser consultada em [Heath97], [Lambert91] ou [Shampine94].

### Métodos explícitos e implícitos

O método de Euler é um método explícito no sentido em que só usa a informação no instante  $t_k$  para avançar a solução para  $t_{k+1}$ . Pode-se usar também informação no instante  $t_{k+1}$ , o que transforma um determinado método em implícito. O exemplo mais simples de um método implícito é o chamado método implícito de Euler (*backward Euler method*):

$$x_{k+1} = x_k + \Delta t_k \times f(x_{k+1}, t_{k+1}) \quad (2.6)$$

Este método é implícito porque existe a necessidade de avaliar a função  $f$  com o argumento  $x_{k+1}$ , antes de se conhecer o seu valor. Isto significa que um valor para  $x_{k+1}$ , que satisfaça a equação anterior, tem que ser determinado. Se  $f$  for uma equação não linear de  $x$ , como é geralmente o caso, é necessário resolver um sistema de equações não lineares em cada passo de integração. Pode então ser usado um método iterativo, como o método de iteração com ponto fixo ou o método de Newton, para a resolução do sistema de equações não lineares. Uma boa estimativa inicial para o processo de iteração pode ser obtido a partir de um método explícito, como o método de Euler, ou através da solução do passo anterior. Uma revisão de métodos numéricos implícitos e explícitos pode consultar-se em [Betts98].

### Sistemas com dinâmicas rápidas e lentas

Uma classe particular de sistemas é caracterizada por apresentar simultaneamente dinâmicas rápidas e lentas. A modelação multi-domínio resulta normalmente neste tipo de sistemas, como, por exemplo, em sistemas mecatrónicos onde um componente mecânico lento é controlado por circuitos eléctricos rápidos ou por actuadores hidráulicos. A modelação matemática nestes casos dá origem a sistemas de equações rígidos (*stiff*) que colocam dificuldades aos algoritmos que implementam métodos explícitos. Para manter a estabilidade do método são necessários passos de integração muito pequenos o que acarreta simulações muito lentas. Formalmente, um ODEs estável é rígido se os valores próprios da matriz Jacobiana  $J$  têm valores muito diferentes. Pode haver um valor próprio com uma parte real muito negativa (correspondendo a um componente da solução

muito amortecida) ou uma parte imaginária grande (correspondendo a um componente da solução com oscilação rápida). Esta equação diferencial pode corresponder a um processo físico cujos componentes possuem diferentes escalas de tempo, ou a um processo em que a escala de tempo é pequena quando comparada com o intervalo em que a solução está a ser avaliada. Na integração deste tipo de sistemas, com a utilização de métodos explícitos, o passo de integração é limitado pelos problemas de instabilidade inerentes ao método. Infelizmente, a constante de tempo mais pequena governa a estabilidade de todo o sistema e, mesmo que um só componente se torne instável, todo o sistema tem que se considerar rígido. Se o passo de integração é grande, então a trajectória começa a oscilar e diverge. A solução standard para os comportamentos instáveis de sistemas rígidos consiste no uso de métodos implícitos. Em resumo, a escolha entre um método explícito implica a integração do sistema de equações com um passo bastante pequeno, de modo a manter a estabilidade, enquanto que a selecção de um método implícito obriga, em cada passo, à resolução de um sistema de equações algébricas, na maioria das vezes de dimensão elevada.

### Solução numérica e inicialização de DAEs

A solução de um DAEs apresenta dificuldades numéricas e analíticas que são diferentes das encontradas na solução de ODEs [Gear84]. De facto, e segundo Aström [Aström98], os algoritmos para a resolução numérica de equações diferenciais algébricas continuam a não estar tão desenvolvidos como aqueles para as equações diferenciais ordinárias. Podem não existir sempre soluções para os DAEs, ou a existirem podem não ser únicas. O interesse por este problema matemático tem vindo a aumentar nos últimos anos; alguns métodos geralmente usados em ODEs, nomeadamente a chamada *backward difference formula* (BDF) e os métodos Runge-Kutta implícitos, têm vindo também a ser usados em DAEs. A forma de resolução de um DAEs consiste, na sua essência, na aproximação de  $x$  por uma fórmula de diferença finitas, ou métodos de Runge-Kutta implícitos, e na resolução do sistema de equações não lineares resultante, através de métodos iterativos, para obter uma aproximação para  $x$ . Por exemplo, o método implícito de Euler ( $x_{k+1} = x_k + h_k \times x_{k+1}$ ) pode ser usado na equação (2.4) obtendo-se assim um sistema de equações algébricas não lineares:

$$\begin{aligned}
 h \frac{d}{dt} (x_{k+1} - x_k), x_{k+1}, y_{k+1}, t_{k+1} \frac{\ddot{\theta}}{\theta} &= 0 \\
 g(x_{k+1}, y_{k+1}, t_{k+1}) &= 0,
 \end{aligned}
 \tag{2.7}$$

O primeiro método para a resolução de DAEs foi proposto por Gear [Gear71]. Existem também alguns métodos que resolvem DAEs directamente [Brenan88].

## 2.3 Sistemas híbridos

Todos os sistemas físicos, ao nível em que a análise é realizada neste trabalho, são pela sua natureza sistemas contínuos. Contudo, existem muitas vezes não linearidades que tornam o seu comportamento complexo e difícil de analisar. A complexidade é por vezes reduzida através de linearizações em torno de pontos particulares de funcionamento do sistema ou através da abstracção da escala temporal para a descrição de alguns comportamentos. Os sistemas de engenharia representam um caso particular dos sistemas físicos, pois são concebidos pelo homem; este tipo de sistemas pode ser visto como uma configuração de elementos físicos interligados, que podem ter dinâmicas muito diversas. Por exemplo, nos sistemas eléctricos podem coexistir elementos de comutação rápida como os díodos, tiristors ou relés combinados com componentes contínuos como bobinas, condensadores e motores eléctricos. Em modelos muito detalhados, até as transições dos elementos de comutação podem ser modelados como transições contínuas; o modelo resultante seria então um sistema rígido de equações contínuas. Num modelo mais prático, as transições dos componentes de comutação rápida podem ser consideradas transições instantâneas e ideais. Noutros domínios podem também ocorrer frequentemente situações idênticas; por exemplo, nos sistemas de transmissão dum veículo automóvel as transições entre as diferentes relações da caixa de velocidades podem ser consideradas transições abruptas ou num sistema hidráulico onde coexistam válvulas proporcionais e válvulas *on-off*, estas podem considerar-se como componentes com transições instantâneas. Outras situações frequentes nos sistemas de engenharia ocorrem quando sistemas contínuos são controlados por algoritmos de controlo implementados em computadores digitais. De facto, os comportamentos híbridos (contínuos e discretos) podem ocorrer naturalmente em sistemas físicos com controladores digitais embebidos, pois o controlador pode forçar o sistema a operar em variadas configurações ou modos; por exemplo, o sistema de *fly-by-wire* do Airbus A-320 inclui os modos de operação *take-off*, *cruise* e *go-around* [Sweet95]. Dentro de cada um dos modos, o comportamento do sistema evolui continuamente, mas as variações discretas nos modos, definidas pelo controlador de supervisão, podem ocorrer em determinados instantes, resultando em descontinuidades no comportamento global do sistema. A modelação de sistemas mecânicos também envolve muitas vezes a modelação de colisões entre corpos rígidos; contudo, a utilização de modelos de impacto bastante precisos, que levam em conta a elasticidade do material, descrevendo o processo de colisão através de modelos contínuos complexos, é inviável, por exemplo, para simulação em tempo real. Além disso, esses modelos são difíceis de construir e analisar pois, mesmo que alguns desses fenómenos pudessem ser descritos com fiabilidade, a determinação precisa dos parâmetros desses modelos seria também uma tarefa

bastante complexa. Estes factos levaram ao uso do conceito dos sistemas híbridos na modelação do impacto em sistemas mecânicos [Bullo98].

A modelação de algumas dinâmicas rápidas através de variações instantâneas, representando o comportamento dinâmico através de modos contínuos intercalados por transições instantâneas, tem vindo a ser usada para a redução da complexidade dos modelos [Mosterman95]. Estas variações nos modos contínuos ocorrem quando as variáveis do sistema passam alguns patamares e quando ocorrem eventos de estado ou eventos de controlo externos. Os modelos híbridos são então uma representação natural para sistemas embebidos (sistemas físicos com controladores digitais) e para sistemas físicos complexos cujo comportamento é simplificado pela introdução de transições discretas em substituição de dinâmicas não lineares rápidas [Mosterman00].

### **2.3.1 Modelos para sistemas híbridos**

O trabalho pioneiro na modelação de sistemas híbridos foi introduzido por Fahrland [Fahrland70]. A modelação, simulação, análise e controlo de sistemas que englobam dinâmica contínua e discreta, apelidados de sistemas híbridos, tem vindo a ser objecto de estudo nos últimos anos a avaliar pelos trabalhos apresentados em [Praehofer91], [Anderson94], [Barton94], [Branicky95], [Sastry95], [Cassandras97], [Schaft98], [Tomlin98], [Fábian99], [Mosterman99], [Beek00] ou [Mosterman00]. A combinação de sistemas contínuos e discretos torna os sistemas híbridos inerentemente difíceis de analisar formalmente. De facto, e segundo Lygeros [Lygeros94], devido à inexistência de ferramentas teóricas a simulação tem um papel muito importante (se não fundamental) no projecto de sistemas híbridos complexos e na sua análise. A formulação dos modelos pode ser realizada em tempo contínuo ou em tempo discreto. Uma formulação em tempo contínuo assume que pelo menos parte do estado do sistema pode ser definido através de funções diferenciais em intervalos de tempo de interesse. Segundo Barton [Barton01] um sistema híbrido pode ser descrito por um conjunto de sub-sistemas de estados discretos, por um conjunto de sub-sistemas de estados contínuos e as possíveis interações entre os dois subsistemas. Na maioria dos trabalhos realizados no âmbito dos sistemas híbridos são usados DAEs ou ODEs, muito embora possam ser usados outros tipos de descrições como os sistemas de equações com derivadas parciais (PDEs). Da mesma forma poderão ser usados diferentes formalismos para os sub-sistemas discretos como, por exemplo, as máquinas de estados finitas [Alur95], as redes de Petri [Mosterman98] ou os Statecharts [Ferreira99a], [Otter00b].

Ao longo dos últimos anos foram propostos vários modelos matemáticos para representação de sistemas híbridos (ver [Branicky98]). O modelo proposto em [Anderson94], baseado no

formalismo do Automata Híbrido [Alur95], é usado de seguida para introduzir a problemática da simulação de sistemas híbridos.

Segundo Anderson um modelo híbrido pode ser representado por uma *tuple*:

$$M = \langle Q, X, E, g, h, f, D \rangle \quad (2.8)$$

onde os componentes da *tuple* são definidos da forma seguinte:

- $Q = (q_1, \dots, q_{n_q})$  é um conjunto de variáveis que representam o estado discreto do sistema. Podem ser variáveis reais, inteiras ou do tipo simbólico. Estes estados são por vezes chamados locais.
- $X = (x_1, \dots, x_{n_x})$  é um conjunto de variáveis reais que representam o estado contínuo do sistema.
- $E = (e_1, \dots, e_{n_e})$  é um conjunto de tipos de eventos.
- $g = (g_1, \dots, g_{n_g})$  é um conjunto de funções que definem o comportamento contínuo pelas equações  $g_i(X(t), \dot{X}(t), Q_t) = 0$ ,  $g_i \hat{=} g$ . A notação  $Q_t$  indica que o estado discreto  $Q$  é actualizado em instantes discretos.
- $h = (h_1, \dots, h_{n_h})$  é um conjunto de funções do tipo booleano, de tal forma que  $h_i(X(t), \dot{X}(t), Q_t) = true$ ,  $h_i \hat{=} h$  define os invariantes do modelo. As funções invariantes dividem o espaço de estados em duas partes: o conjunto de estados admissíveis onde todas as  $h_i$  tem valor lógico *true* e os restantes estados onde pelo menos uma dessas funções tem o valor *false*.
- $f$  é o mapa  $h \otimes E$  que associa cada função invariante a um tipo de evento.
- $D = (d_1, \dots, d_{n_e})$  é o conjunto de funções vectoriais, cada uma associada a um tipo de evento,  $e_i \hat{=} E$ , tal que quando  $e_i$  ocorre, então a equação  $d_i(X_a, \dot{X}_a, Q_a, X_b, \dot{X}_b, Q_b) = 0$  é satisfeita.  $X_a$  e  $Q_a$  referem-se ao estado imediatamente após o evento, enquanto  $X_b$  e  $Q_b$  referem-se ao estado imediatamente antes do evento.

O modelo híbrido acima descrito representa um sistema cujo estado evolui de forma contínua no tempo, através da variação das variáveis  $X$ , e de forma discreta, através de variações instantâneas no estado total representado pelas variáveis  $Q$  e  $X$ . Estas mudanças instantâneas, designadas por eventos, ocorrem quando o sistema abandona o conjunto de estados admissíveis, definidos pelas funções invariantes.

O conjunto de funções  $g$  representa um DAEs que define o comportamento contínuo do sistema e pode incluir constrangimentos algébricos entre variáveis discretas. O estado discreto  $Q$  é assumido constante no intervalo de tempo entre eventos, isto é, tem derivada nula no que diz respeito ao comportamento contínuo, só podendo variar quando ocorre um evento. Os eventos ocorrem de forma assíncrona e são gerados quando um invariante é violado (quando qualquer  $h_i(t)$  varia de *true* para *false*) de acordo com o mapa  $\phi$ . O disparo de um evento pode provocar uma transição descontínua no estado; essa transição é definida pelo conjunto de funções em  $D$  associadas com o tipo de evento detectado. Contudo, o disparo de um evento pode causar a violação de outros invariantes; neste caso, os eventos devem ser disparados até que se atinja uma situação estável onde todos os invariantes sejam respeitados. No caso em que existe habilitação de eventos em simultâneo, num mesmo instante, estes podem ser disparados de uma forma determinística ou não determinística. Uma forma de ordenar o disparo de eventos é a atribuição de prioridades aos eventos.

A evolução do sistema híbrido é então uma sequência de estados que respeita todas as restrições. Um exemplo de uma evolução está representada na Figura 2.3: o sistema híbrido tem duas variáveis de estado  $x_1$  e  $x_2$ ; estão representados três locais ( $s_1, s_2, s_3$ ) com os seus conjuntos de invariantes e as actividades possíveis; a evolução inicia-se no estado contínuo  $s_0$  e termina no estado  $s_1$ .

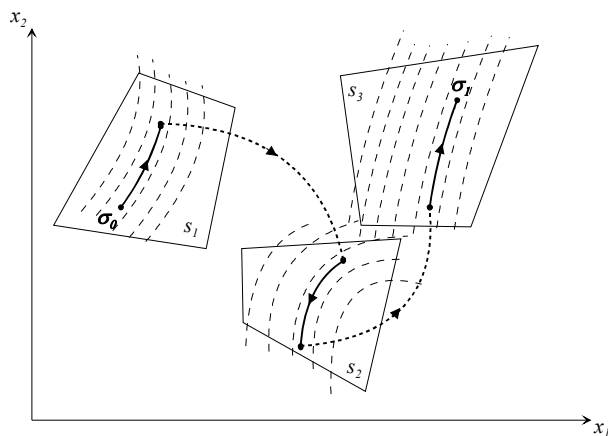


Figura 2.3 Comportamento multi-modo nos sistemas híbridos

### 2.3.2 Simulação de modelos híbridos

Para a simulação de um modelo híbrido é necessário um conjunto de condições iniciais. Estas condições iniciais são constrangimentos adicionais das variáveis do modelo e das suas derivadas no instante inicial  $t_0$ . Um problema de simulação está bem definido se as condições iniciais são



consistentes e determinam valores iniciais para todas as variáveis. Além disso, o próprio modelo deve estar correctamente definido para fornecer uma solução única durante o tempo de simulação. As condições iniciais podem ser definidas pelo utilizador, no ambiente de simulação, ou no próprio modelo. Pode-se então definir o tempo inicial  $t_0$  e os valores iniciais de algumas variáveis contínuas, derivadas ou variáveis discretas de tal forma que a equação  $g(x_0, \dot{x}, Q_0) = 0$  possa ser resolvida para todos os outros valores iniciais desconhecidos. Assumindo valores iniciais consistentes, a simulação de um modelo híbrido, como o definido anteriormente, pode ser realizada pelo algoritmo da Figura 2.4.

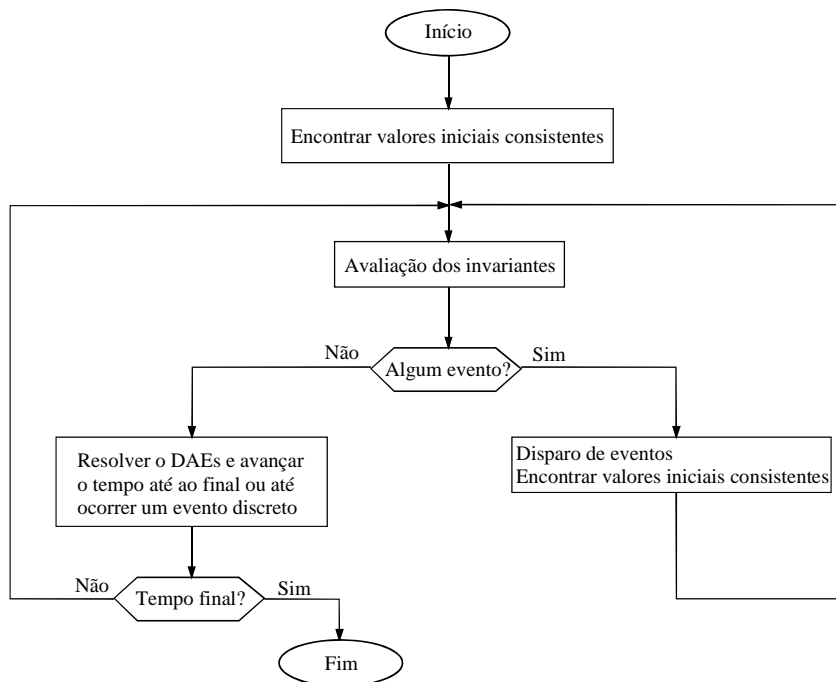


Figura 2.4 Fluxograma dum algoritmo usado na simulação de modelos híbridos

O problema de detecção de eventos implica que os invariantes sejam continuamente avaliados durante a integração numérica do DAEs de modo a detectar quando pelo menos uma das funções invariantes se torna falsa. Um elemento chave para a simulação de sistemas híbridos é, de facto, o mecanismo para a detecção precisa e efectiva dos instantes em que os invariantes são violados. Isto pode ser crítico, pois uma sequência de eventos errada pode levar a trajectórias muito diferentes para o estado contínuo. As rotinas de integração standard não estão habilitadas a detectar eficientemente os eventos de estado. Em princípio, a detecção da violação dos invariantes deve ser efectuada através de técnicas de recuo do passo (*backstepping*) com o modelo válido antes do evento de estado ocorrer. Este mecanismo de *backstepping*, para detecção mais precisa dos instantes em que ocorrem os eventos, pode ser realizado nos métodos numéricos de passo variável através da redução do passo de integração.

O disparo de um evento significa a determinação das variáveis afectadas pelo evento e o cálculo dos seus novos valores, os quais, sendo o estado inicial do novo DAEs, devem ser consistentes com as equações que determinam o comportamento contínuo. A execução de um evento pode ser vista, deste modo, como um problema de re-inicialização.

Pode-se então considerar que estamos perante dois modos distintos de simulação: a) a simulação em tempo contínuo, em que se assume todas as variáveis discretas como contínuas e se resolve o sistema de equações contínuas no tempo; entretanto os indicadores de eventos (os invariantes) são continuamente avaliados de modo a mudar o modo de simulação caso ocorra um evento; b) a simulação de eventos discretos, onde o evento é seleccionado e onde são executadas todas as acções associadas ao evento, incluindo o disparo de novos eventos e a realização das acções correspondentes; o processo é repetido até que não ocorram mais eventos, isto é, se atinja uma situação estável, após a qual o modo de simulação contínuo é de novo activado.

A passagem do modo de simulação discreto para a simulação em tempo contínuo implica a re-inicialização do novo DAEs. A reinicialização nem sempre é de resolução trivial pois pode ser um problema não linear. De notar que toda a simulação discreta ocorre em tempo nulo do ponto de vista da simulação contínua. Este facto está de acordo com o estabelecido por Kapur, em [Kapur97], quando este refere que os sistemas híbridos podem ser vistos como sistemas de tempo real que permitem mudanças contínuas no estado, durante certos períodos de tempo, e mudanças discretas em tempo nulo.

### **2.3.3 Exemplo de um sistema híbrido: servomecanismo hidráulico**

Esta secção apresenta a modelação de um servomecanismo hidráulico como um exemplo de modelação de sistemas físicos que envolvem vários domínios, bem como componentes com comportamentos contínuos e/ou discretos no tempo.

Os actuadores angulares de palheta são muitas vezes utilizados em manipuladores para a realização de movimentos angulares. O esquema normalmente usado está representado na Figura 2.5 e pode descrever-se sucintamente da seguinte forma: a posição angular é controlada em malha fechada por um controlador de posição; este controlador tem como entrada uma referência de posição angular e a posição real (normalmente fornecida por um codificador angular) e, como saída, fornece um sinal de controlo a uma válvula proporcional responsável pelo comando do actuador.

O sistema é composto por um conjunto de componentes interligados entre si através de tubagens, no caso das ligações entre componentes hidráulicos, ou outro tipo de ligações, como, por exemplo, no caso do transdutor de posição e o veio do actuador angular. O objectivo é controlar em malha

fechada, através de um controlador implementado discretamente, a posição angular de um veio acoplado a um actuador angular hidráulico. O comando do actuador é feito através de uma válvula de controlo direcciona em que o movimento da gaveta é controlado em malha fechada por forma a que a dinâmica do movimento da gaveta seja tal que, face à dinâmica global, possa desprezar-se. As válvulas de segurança e as válvulas de retenção limitam a aceleração nos dois sentidos.

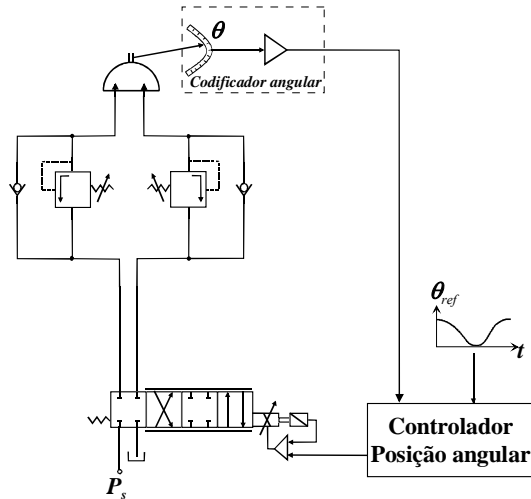


Figura 2.5 Servomecanismo hidráulico de actuação angular

Este exemplo simples evidencia algumas das dificuldades na modelação deste tipo de sistemas, onde se incluem domínios físicos vários e a coexistência de modelos discretos e contínuos para a modelação do sistema como um todo. Os modelos de alguns componentes resultam da aplicação de leis fundamentais da mecânica dos fluidos; outros, devido à menor importância no desempenho global, resultam de aproximações a comportamentos de dispositivos ideais; outros ainda são simplificados modelando somente o seu comportamento global.

O modelo do servomecanismo hidráulico pode então ser descrito pelos seus componentes interligados por interfaces adequadas. Analisando o esquema da Figura 2.5 verifica-se a existência de componentes repetidos, como são as válvulas de retenção e as válvulas de segurança e também outros quatro componentes: actuador angular, servoválvula proporcional, codificador angular e um controlador de posição.

### Modelo matemático do actuador angular

O diagrama do actuador angular de palheta é apresentado na Figura 2.6. O binário de saída do actuador, assumindo eficiência de 100%, é dado por:

$$t_{out} = D(P_1 - P_2) \quad (2.9)$$

onde  $D$  é o volume por unidade de ângulo (cilindrada do actuador).

As equações diferenciais que descrevem a evolução no tempo das pressões nas câmaras do actuador resultam da aplicação da equação de continuidade [Merrit67] a ambas as câmaras do actuador. Considerando que não existem fugas externas e assumindo que as fugas internas são laminares, representadas por uma condutância de fugas,  $g_{lk}$ , tem-se:

$$\frac{dP_1}{dt} \frac{V_1}{\beta_e} + \frac{dV_1}{dt} = Q_1 - g_{lk} (P_1 - P_2) \quad (2.10)$$

$$\frac{dP_2}{dt} \frac{V_2}{\beta_e} + \frac{dV_2}{dt} = -Q_2 + g_{lk} (P_1 - P_2) \quad (2.11)$$

$\beta_e$  é o módulo de compressibilidade efectivo do sistema (incluindo o óleo, tubagens,...) e  $V_1$  e  $V_2$  são os volumes totais nas câmaras do actuador, incluindo tubagens e ligações, isto é:

$$V_1 = V_{L1} + D \times q \quad (2.12)$$

$$V_2 = V_{L2} + D \times (q_{\max} - q) \quad (2.13)$$

em que  $V_{L1}, V_{L2}$  representam os volumes “mortos” das câmaras 1 e 2 do actuador (volumes das condutas, câmaras da válvula, ...) e  $q_{\max}$  é o curso máximo do actudor em radianos. Como a variação de volume nas duas câmaras é simétrica, como se depreende da Figura 2.6, então:

$$\frac{dV_1}{dt} = - \frac{dV_2}{dt} = D \frac{dq}{dt} = D \times w \quad (2.14)$$

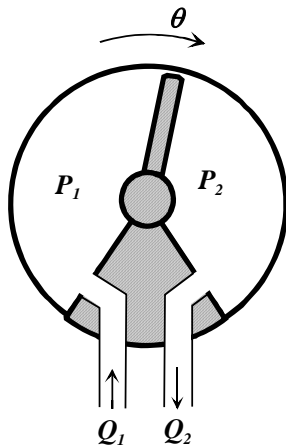


Figura 2.6 Actuador de palheta

Se se considerar que a inércia total em movimento é  $J$ , e que o sistema tem uma componente de atrito de Coulomb  $f_{co}$  e um coeficiente de atrito viscoso  $k_v$ , a aplicação da 2ª lei de Newton resulta em:

$$J \frac{d^2q}{dt^2} = t - f_{co} \times \text{sign} \langle w \rangle - k_v \times w \quad (2.15)$$

## Modelo da servoválvula proporcional

Na Figura 2.7 é apresentado um diagrama de uma válvula proporcional de gaveta de centro crítico.

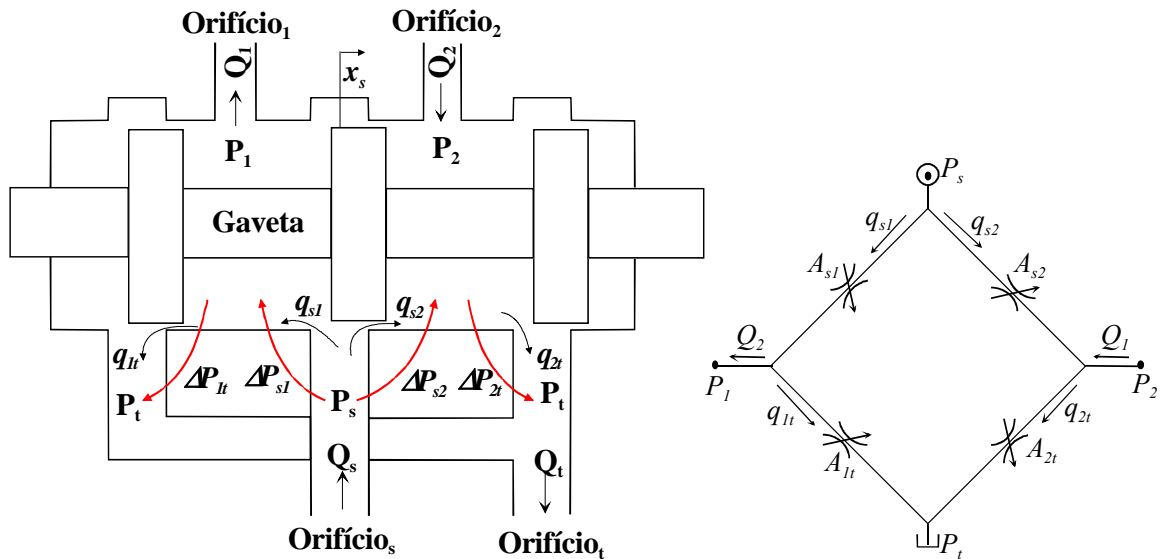


Figura 2.7 Diagrama de uma válvula proporcional e sua representação em ponte de Wheatstone

Os caudais volumétricos através dos orifícios da válvula são definidos pelas secções de passagem que, por sua vez, são função da posição da gaveta da válvula,  $x_s$ . Uma das simplificações normalmente usada considera os quatro orifícios emparelhados e simétricos, isto é:

$$A_{s1}(x_s) = A_{t2}(x_s) = A_{s2}(-x_s) = A_{t1}(-x_s) \quad (2.16)$$

Considerando que a secção de cada orifício varia proporcionalmente com a posição da gaveta, isto é:  $A_{ij}(x_s) = k \times x_s$ , onde  $k$  é o ganho de área do orifício, e assumindo que o escoamento é sempre turbulento, a equação de orifício [Merrit67] pode ser usada para descrever os caudais em função da posição da gaveta e da diferença de pressão entre os dois lados do orifício:

$$q = k \times x_s \times C_d \sqrt{\frac{2DP}{\rho}} \quad (2.17)$$

onde  $C_d$  é o coeficiente de descarga do orifício,  $\rho$  é a massa específica e  $\Delta P$  é a diferença de pressão.

Desta forma, os caudais volumétricos  $Q_1$  e  $Q_2$  podem ser calculados de forma estática:

$$\begin{aligned} \ddagger Q_1 &= q_{s1} - q_{t1} \\ \ddagger Q_2 &= q_{t2} - q_{s2} \end{aligned} \quad (2.18)$$

$$\begin{aligned}
q_{s1} &= \text{sign}(DP_{s1})k \times x_s \sqrt{\frac{2|DP_{s1}|}{r}} && ;x_s \geq 0 \\
q_{s1} &= 0 && ;x_s < 0 \\
q_{s2} &= \text{sign}(DP_{s2})k \times x_s \sqrt{\frac{2|DP_{s2}|}{r}} && ;x_s \leq 0 \\
q_{s2} &= 0 && ;x_s > 0 \\
q_{1t} &= \text{sign}(DP_{1t})k \times x_s \sqrt{\frac{2|DP_{1t}|}{r}} && ;x_s \leq 0 \\
q_{1t} &= 0 && ;x_s > 0 \\
q_{2t} &= \text{sign}(DP_{2t})k \times x_s \sqrt{\frac{2|DP_{2t}|}{r}} && ;x_s \geq 0 \\
q_{2t} &= 0 && ;x_s < 0
\end{aligned} \tag{2.19}$$

Os efeitos das forças de caudal e da histerese, presente no movimento da gaveta neste tipo de válvulas, são minimizados pelo controlo em malha fechada do movimento da gaveta, de tal forma que a válvula pode ser descrita por um modelo estático simples, muito embora com descontinuidades na primeira derivada das equações dos caudais na posição central da gaveta.

### Modelação do codificador angular

O codificador angular pode ser modelado tendo em conta a sua resolução. Por exemplo, um codificador de 5000 posições possíveis em  $2\pi$  radianos poder-se-á modelar através de um bloco algébrico com uma entrada e uma saída:

$$q_k = \text{round}\left(\frac{5000 \times q}{2p}\right) \times \frac{2p}{5000} \tag{2.20}$$

### Modelação da válvula de segurança

As válvulas de segurança são usadas para limitar as pressões nos sistemas, muito embora também possam ser usadas como válvulas de contra pressão como no exemplo presente. São dispositivos que funcionam em malha fechada, mas são normalmente modelados através das suas características de entrada/saída estáticas [Beater98]. Uma análise pormenorizada pode ser vista em [Merrit67].

Uma análise simples, do comportamento da válvula de segurança, consiste em representar a válvula através de uma máquina de estados:

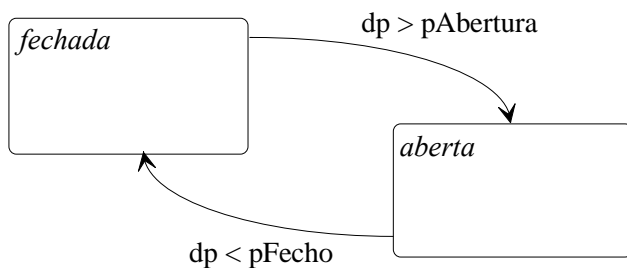


Figura 2.8 Máquina de estados para uma válvula de segurança

A máquina de estados da Figura 2.8 pode ser implementada da seguinte forma:

$$\begin{aligned} fechada(t) &= (fechada(t^-) \wedge dp(t) \leq pAbertura) \vee (aberta(t^-) \wedge dp(t) < pFecho) \\ aberta(t) &= (aberta(t^-) \wedge dp(t) \geq pFecho) \vee (fechada(t^-) \wedge dp(t) > pAbertura) \end{aligned} \quad (2.21)$$

onde  $t^-$  significa o instante imediatamente anterior à avaliação da expressão booleana. O caudal  $q_r$  depende do estado da válvula (aberta ou fechada):

$$\begin{aligned} q_r &= DP \times g_{lk}; fechada \\ q_r &= (DP - pFecho) \times g_{op} + DP \times g_{lk}; aberta \end{aligned} \quad (2.22)$$

$g_{lk}$  e  $g_{op}$  representam as condutâncias de fugas e de válvula aberta, respectivamente.

### Modelação do controlador discreto

Supondo a utilização de um controlador PI (Proporcional + Integral) discreto, o controlador poderia ser implementado pelo algoritmo seguinte, avaliado em intervalos de tempo  $T_s$ :

$$\begin{aligned} erro(t) &:= q_{ref}(t) - q(t) \\ x(t) &:= x(t - T_s) + erro(t) \times \frac{T_s}{T_I} \\ PI_{out}(t) &:= K \times (x(t) + erro(t)) \end{aligned} \quad (2.23)$$

Onde  $T_I$  é a constante de tempo integral do controlador,  $K$  representa o ganho,  $x$  o estado, sendo  $erro$  a diferença entre a posição angular de referência e real do actuador.

## 2.4 Linguagens de modelação e simulação

Os primeiros simuladores foram implementados em hardware e de forma analógica. A ideia principal era a realização de um dispositivo físico que resolvesse as equações diferenciais ordinárias que modelavam o sistema. A primeira ferramenta para simular sistemas dinâmicos mecânicos foi desenvolvido no MIT e relatado em [Bush31]. Com o aparecimento dos computadores analógicos, nos anos 50, as equações diferenciais do sistema eram transformadas num formato de espaço de estados de modo a estabelecer os diagramas a introduzir no computador analógico. Devido à disponibilidade para execução de operações matemáticas em paralelo, estes computadores eram especialmente importantes para realizar simulações para a indústria aeronáutica e aeroespacial.

Quando surgiram os computadores digitais houve uma tendência para explorar a sua utilização no desenvolvimento de simuladores. No princípio dos anos 60 foi desenvolvida uma grande variedade de pacotes de software para a simulação digital de sistemas dinâmicos contínuos. Esses produtos

eram geralmente codificados em linguagens de baixo nível, ou em FORTRAN, ou então continham elementos que dependiam da arquitetura do sistema de hardware, impedindo a sua portabilidade entre diferentes máquinas ou compiladores. Estas técnicas necessitavam de muito tempo de desenvolvimento e conhecimentos em métodos numéricos e programação, para além do conhecimento dos processos físicos necessário ao desenvolvimento dos modelos, o que causava longos períodos de familiarização para se poder fazer modificações no ambiente de simulação. Esta situação levou ao aparecimento de um standard denominado CSSL (*Continuous System Simulation Language*) [Strauss67], segundo o qual um programa de simulação digital pode ser dividido em três partes: Inicial, Dinâmica e Terminal. A secção inicial contém os cálculos que somente necessitam de ser executados uma vez por simulação. A parte de simulação dinâmica contém os cálculos e as operações de entrada/saída que devem ser realizadas em cada incremento da variável independente, também muitas vezes chamado intervalo de comunicação; este não pode ser confundido com o intervalo ou passo de integração que é determinado pela precisão desejada e pelo algoritmo de integração utilizado. A região terminal da simulação recebe o controlo do programa que implementa a parte dinâmica, de modo a executar as tarefas necessárias para terminar a simulação; podem ainda ser executadas tarefas de armazenamento e de entrada/saída. Uma perspectiva geral sobre o standard CSSL é apresentado em [Rimvall86].

### **Linguagens procedimentais, declarativas e diagramas de blocos**

Muitas das linguagens que foram baseadas no standard CSSL são procedimentais por natureza. Os modelos são desenvolvidos à custa de instruções de atribuição, à semelhança do que acontece nas linguagens de programação. As atribuições especificam variáveis dependentes como funções de variáveis independentes (causalidade fixa) e devem ser executadas na sequência definida pelo modelador.

As linguagens declarativas, também conhecidas como linguagens baseadas em equações, não impõem causalidade ao modelo, sendo este definido por um conjunto de equações que estabelecem relações entre os estados, as suas derivadas e o tempo. Estas linguagens permitem a especificação directa das equações diferenciais, com ordem arbitrária, usando os nomes das variáveis definidas pelo utilizador. Às linguagens declarativas está associado um simulador responsável por converter essas equações em instruções com a sequência adequada.

Os diagramas de blocos representam uma forma de especificação gráfica de modelos e têm especial interesse na área do controlo. São compostos por blocos ligados por setas; as setas especificam o sentido do fluxo de informação e ligam saídas de blocos a entradas de blocos. Assim, os diagramas de blocos são causais por natureza. A modelação é essencialmente gráfica, podendo os diagramas



de blocos ser especificados de forma hierárquica. Os simuladores que usam diagramas de blocos oferecem um editor para diagramas de blocos, blocos para simulação em tempo contínuo (equações diferenciais), blocos para simulação em tempo discreto (equações de diferenças) e blocos compostos.

### **Linguagens de domínio específico e linguagens gerais**

Algumas das linguagens de modelação são genéricas, aplicáveis às mais variadas áreas, outras são dedicadas, apresentando particularidades que as tornam bastante potentes e eficazes na modelação e simulação nas áreas para as quais foram criadas.

Existe um grande número de pacotes de simulação especialmente desenvolvidos para áreas específicas. Esses pacotes, em geral, disponibilizam um conjunto de modelos agrupados em bibliotecas e ambientes gráficos que permitem compor modelos mais complexos e desenvolver novas bibliotecas de modelos. Por exemplo, para a modelação da dinâmica de sistemas mecânicos, algumas aplicações geram automaticamente as equações do movimento através de descrições geométricas dos mecanismos. É o caso do ADAMS ([www.adams.com](http://www.adams.com) 2003) que pode ser integrado com software de CAD, de tal forma que a geometria do mecanismo pode ser extraída a partir do modelo de CAD. Para sistemas eléctricos existem várias linguagens de simulação, das quais se destacam a PSPICE ([www.pspice.com](http://www.pspice.com) 2003) para circuitos analógicos, e o VHDL e Verilog ([www.accellera.org](http://www.accellera.org) 2003) para circuitos digitais. Mais recentemente encontra-se em desenvolvimento a linguagem VHDL-AMS ([www.vhdl-ams.com](http://www.vhdl-ams.com) 2003) para a simulação de sistemas mistos, analógicos e digitais e para sistemas multi-disciplinares. Para a modelação de sistemas hidráulicos e fluídicos o pacote Amesim ([www.amesim.com](http://www.amesim.com) 2003) fornece um conjunto de bibliotecas de componentes pré-compilados (hidráulicos, pneumáticos, mecânicos, ...).

Um segundo grupo inclui pacotes de simulação que podem ser usados em domínios multi-disciplinares. Um dos exemplos de maior sucesso nas linguagens procedimentais é o ACSL (*Advanced Continuous Simulation Language*) [Mitchell76]. Foi introduzida como uma ferramenta para analisar o comportamento de sistemas dinâmicos descritos por equações diferenciais não-lineares dependentes do tempo e é, ainda hoje, uma das linguagens de modelação mais utilizadas. Emprega um pre-processamento para FORTRAN, proporcionando independência da plataforma de hardware para simulação dos modelos, o que resulta numa grande eficiência do código gerado. Outras plataformas baseadas em diagramas de blocos, como o Simulink ([www.mathworks.com](http://www.mathworks.com) 2003) ou Easy5 ([www.boeing.com](http://www.boeing.com) 2003), fornecem ambientes interactivos para modelação de sistemas multi-disciplinares.

## **Linguagens orientadas por objectos**

O paradigma da modelação orientada por objectos, introduzido por Elmqvist no fim dos anos 70 [Elmqvist78], levou ao desenvolvimento de um grande número de linguagens de modelação que implementam alguns dos conceitos das linguagens orientadas por objectos, de modo a facilitar a reutilização e manutenção de bibliotecas de modelos. Cellier, em [Cellier96], descreve as características que um ambiente de modelação orientado por objectos para a modelação de sistemas físicos deve fornecer: o encapsulamento, como a possibilidade de codificar todo o conhecimento relacionado com um determinado componente de uma forma compacta e com interface bem definida com o exterior; a capacidade de interligação de componentes de modo a definir topologias de organização, o que requer que as equações que descrevem o comportamento do modelo sejam declarativas para que o modelo seja não causal; a modelação hierárquica, de modo a permitir a declaração de modelos interligados como novos objectos, não sendo distinguíveis do exterior de outros modelos mais simples; a instanciação de objectos, de modo a que se possam definir classes de objectos genéricos e instanciar objectos das definições de classes através de mecanismos de invocação de classes; a herança de classes, que permita o encapsulamento do conhecimento de um dado objecto físico, de modo a que este possa ser distribuído através de mecanismos de herança, assegurando que um mesmo conhecimento do modelo não tem que ser codificado várias vezes em diferentes partes do modelo; a capacidade de interligação, de modo a interligar modelos de componentes através de nodos. Estes nodos são diferentes dos outros modelos de objectos, já que devem permitir um número variável de ligações. Nos modelos de sistemas físicos é necessária a disponibilidade de dois tipos de variáveis nos nodos, para que a continuidade de potência através dos nodos seja assegurada.

Ao longo das duas últimas décadas estes conceitos foram implementados em várias linguagens de modelação orientadas por objectos. À pioneira Dymola [Elmqvist78] seguiram-se as linguagens ASCEND [Piela91], gProms [Barton94], Omola [Anderson94], Smile [Kloas95], ObjectMath [Viklund95], NMF [Sahlin96] ou ULM [Jeandel96]. Esta situação era idêntica à que ocorreu em meados dos anos 60 quando o standard CSSL surgiu de numa necessidade de uniformizar a grande diversidade de programas de simulação. Recentemente a linguagem VHDL-AMS foi standardizada pelo IEEE para a modelação de sistemas com componentes analógicos e digitais. Uma outra linguagem, chamada Modelica [Elmqvist97], surgiu de um esforço internacional iniciado na Europa em 1996 e tendo como objectivo desenvolver um formato standard que permitisse a troca de modelos de diferentes domínios entre diferentes utilizadores ou ferramentas.

A linguagem Modelica, que será abordada adiante em secção própria, é uma linguagem de modelação orientada por objectos de uso geral, que tem sido desenvolvida com o objectivo de definir uma linguagem “unificada” para a modelação de sistemas físicos. A linguagem Modelica suporta a modelação orientada por objectos através do uso de conceitos como a herança, introduzidos pelas linguagens como o Simula [Kirkerud89], Eiffel [Meyer92] ou C++ [Stroustrup85], de modo a facilitar a estruturação dos modelos. Também suporta modelação não causal, não necessitando que sejam atribuídas entradas e saídas aos modelos na fase de modelação. Isto simplifica as tarefas de modelação e reutilização dos modelos já que cada componente pode ser modelado de forma independente. As entradas e saídas do modelo são determinadas pelo ambiente onde o modelo está inserido, sendo explicitadas pela ferramenta de simulação quando o modelo do sistema é compilado. Uma das características principais da linguagem é a possibilidade de componentes pertencentes a bibliotecas de domínios diferentes poderem ser ligados entre si. Um dos maiores problemas das ferramentas de modelação e simulação de domínios específicos tem a ver com a dificuldade de simulação dos modelos noutros simuladores; a transferência de modelos entre ferramentas pode envolver a conversão de ficheiros de dados, ou até novo ajuste de parâmetros, para que sejam obtidos os mesmos resultados. A linguagem Modelica vai ser brevemente proposta como um standard ISO e poderá então ser usada para o intercâmbio de modelos entre diferentes ferramentas de simulação, desde que estas ferramentas interpretem o standard Modelica.

### **Classificação das linguagens de modelação**

A classificação mais utilizada dos modelos e linguagens de modelação é baseada na relação entre as mudanças de estado e a evolução no tempo dos modelos. Estas relações são geralmente divididas em três categorias: linguagens para sistemas contínuos no tempo (CT), linguagens de eventos discretos (DE) e linguagens de sistemas discretos no tempo (DT). Contudo, esta última categoria pode ser considerada englobada na anterior onde os eventos ocorrem em instantes pré-determinados. Beek, em [Beek00], propõe o agrupamento em cinco categorias: CT, CT+, DE, DE+ e CT/DE. As linguagens CT são adequadas à modelação de sistemas contínuos no tempo, as linguagens DE são adequadas à especificação de modelos de eventos discretos. As linguagens híbridas, segundo Beek, são divididas em três categorias: CT+, DE+ e CT/DE. As linguagens CT+ são linguagens destinadas à modelação de sistemas físicos, através de equações matemáticas com alguns mecanismos de eventos discretos, para a modelação de descontinuidades ou de acções de controlo discreto. Exemplos de linguagens, ou ambientes de simulação, “pertencentes” a esta categoria são as linguagens ACSL, Dymola, Omola, Modelica ou o Simulink. As linguagens DE+

são linguagens DE estendidas com elementos simples para modelar sistemas contínuos como, por exemplo, processos contínuos tais como simples escoamentos entre vasos comunicantes. Exemplos desta categoria são a linguagem de modelação SIMAN [Pedgen95] ou o Simple++ [Geuder95]. As linguagens CT/DE fornecem elementos de alto nível para a modelação de sistemas contínuos e/ou de eventos discretos. Shift [Deshpande98], VHDL-AMS ou a linguagem Chi [Beek95] são exemplos deste tipo de linguagens.

Em [Otter95], [Aström98], [Mosterman99], [Beek00] e [Sinha01] são apresentadas revisões das linguagens e ambientes de simulação que vêm sendo desenvolvidas desde a década de 60.

## 2.5 Formalismos para descrição de comportamento

Existem duas abordagens principais para a modelação de sistemas de engenharia ou sistemas físicos com dinâmica contínua e discreta: uma corresponde à extensão dos formalismos contínuos com aspectos relacionados com eventos discretos como é o caso dos Bond Graphs, em que são introduzidas estruturas de comutação para a representação de alguns sistemas híbridos; outra aproximação, também abordada nesta secção, corresponde à extensão de formalismos discretos, como as Máquinas de Estados Finitas (FSM) [Kohavi78], redes de Petri [Petri73] ou os Statecharts [Harel87], com dinâmica contínua. O formalismo dos Statecharts, por ser o usado neste trabalho, é abordado com maior detalhe em secção própria.

### Automata híbrido

Um dos formalismos híbridos muitas vezes usado é o Automata Híbrido (HA) [Alur95]. O HA permite a representação de sistemas que envolvem eventos discretos e variações contínuas no tempo, e representa uma evolução do formalismo do Automata de Estados Finitos (FSA), através do acrescento das noções de tempo e de evolução contínua do estado. No HA as transições discretas transportam o sistema entre um número finito de estados de controlo ou locais. Cada local pode ser marcado com as suas leis de evolução contínuas (conjunto de equações diferenciais e algébricas) que definem a forma de variação no tempo de determinadas variáveis, isto é, definem a actividade dentro de um local particular. Os locais são ligados por transições, cada uma marcada com instruções de protecção que permitem habilitar ou inibir a transição. Os locais também estão marcados com invariantes que devem ser respeitados para que o local permaneça activo. Sempre que um dos invariantes não é respeitado é gerada uma excepção responsável pelo disparo de uma

transição. O estado total de um HA é representado por um local e pelo estado contínuo. O estado pode mudar instantaneamente, nos eventos discretos, de acordo com as relações de transição ou pode variar continuamente, através duma variação temporal, de acordo com as actividades definidas nos locais. Os invariantes devem ser assegurados em qualquer instante. Uma evolução do HA é uma sequência de estados que respeita todas as restrições. Uma das variantes do formalismo HA é o HA linear em que os componentes contínuos do sistema só podem variar linearmente. Uma das utilizações do HA linear é a síntese de controladores [Wong-Toi97], [Alur97], [Tittus95].

### **Redes de ligações energéticas híbridas (Bond Graphs híbridos)**

O formalismo dos BG foi inicialmente proposto por Paynter [Paynter61]. As redes de ligações energéticas são essencialmente um grafo da interacção da estrutura de energia num sistema físico. Os BG baseiam-se no princípio da continuidade da potência para a descrição dos sistemas, em vez do princípio da conservação da energia. Como o balanceamento de potências é uma propriedade local em qualquer sistema físico, podem-se aplicar as equações do balanceamento de potências aos diversos sub-sistemas e às suas interfaces e, em seguida, interligar esses subsistemas obtendo-se, dessa forma, a descrição global do sistema. A potência  $P$  em qualquer sistema físico pode ser descrita através de duas variáveis, o esforço e o fluxo, na terminologia dos BG. Os elementos para a construção dos BG são modelos ideais generalizados para todos os domínios físicos e são regidos pelo princípio de conservação de energia; os elementos primitivos são as resistências ou dissipadores ( $\mathbf{R}$ ), elementos armazenadores de energia (capacitivos ( $\mathbf{C}$ ) ou elementos inerciais ( $\mathbf{I}$ ), fontes ideais de esforço ( $\mathbf{S}_e$ ) e de fluxo ( $\mathbf{S}_f$ ) e elementos de transformação (Transformador ( $\mathbf{T}_f$ ) e girador ( $\mathbf{G}_y$ )). Esses elementos têm diferentes formas físicas em domínios diferentes sendo a interacção entre eles independente do domínio. A transferência de energia é representada através de ligações energéticas, em que cada ligação tem associada uma variável de esforço e outra de fluxo, onde *esforço*  $\cdot$  *fluxo* = *potência* é a taxa de variação de energia. Estas ligações são estabelecidas por junções conservadoras de energia que podem ser do tipo fluxo comum (série): junção **1**; ou do tipo esforço comum (paralelo): junção **0**. Estes dois tipos de junções estão presentes em todos os sistemas físicos; por exemplo, num sistema mecânico, a 2ª lei de Newton pode ser formulada por uma junção **1**; num sistema eléctrico a 1ª lei de Kirchoff (lei dos nós) pode ser formulada por uma junção **0**, enquanto a 2ª lei (lei das malhas) pode ser formulada por uma junção **1**.

A introdução de componentes de comutação nos BG deu origem aos chamados BG híbridos. O método mais simples corresponde à introdução de uma junção de um componente de comutação ideal com dois modos: fluxo zero ou esforço zero [Söderman95]. Este componente comuta os dois modos quando a sua condição de transição é verdadeira. Esta aproximação foi usada por Buisson,

em [Buisson97], para modificar a topologia do BG em instantes de comutação. Outra aproximação tem a ver com a utilização de junções controladas [Mosterman97]. Essas junções actuam como junções normais quando estão no estado activo e, quando inactivas, inibem a transferência da energia entre os fragmentos dos modelos que estão ligados pela junção. Nesta aproximação podem ser usados formalismos de eventos discretos, como o FSM, para modelar as variações discretas na configuração do modelo, e o BG para capturar a evolução contínua do comportamento do sistema. A FSM monitoriza o modelo BG e gera sinais de controlo que levam à comutação de algumas junções controladas de *on* para *off* ou vice-versa.

### **Redes de Petri híbridas**

As Redes de Petri (PN) são uma representação gráfica e um formalismo matemático normalmente usado como ferramenta de modelação de sistemas de eventos discretos. Uma das características das PN é a possibilidade de visualizar comportamentos que possuam paralelismo, concorrência, sincronização e partilha de recursos. As PN são compostas por lugares, transições e arcos. Os lugares e as transições são ligados por arcos direccionados de transições para lugares ou vice-versa. Numa das variantes mais usadas, as redes de Petri marcadas, os lugares contêm um número inteiro de marcas, as quais, num determinado instante, definem o estado do sistema modelado pela PN. Assim sendo, a evolução das marcas na rede, devido ao disparo das transições, corresponde à evolução do estado do sistema. O disparo das transições só é possível quando a transição está habilitada, o que acontece quando os lugares de entrada da transição contêm pelo menos uma marca. A consequência do disparo de uma transição é a subtracção de uma marca de cada um dos lugares de entrada e a adição de uma marca a cada lugar de saída da transição. Algumas variantes das redes de Petri foram surgindo ao longo dos anos (para uma revisão consultar [David92]).

Outra das variantes das PN são as PN híbridas. Estas contêm uma parte discreta e uma parte contínua (existência de lugares contínuos e discretos, e transições contínuas e discretas), podendo ser usadas para a modelação do comportamento de sistemas híbridos. Um lugar contínuo é marcado com um número real, em vez de um número inteiro. O disparo contínuo de uma transição origina uma fracção das marcas a abandonar o local. O comportamento de transição da rede é determinado pela chamada velocidade de disparo das transições, que pode ser constante ou depender da marcação dos lugares de entrada na transição (redes com velocidade de disparo constante ou variável). Em algumas implementações esta velocidade pode ou não depender do tempo.

Ao longo dos tempos foram propostas diferentes formas onde as PN são usadas para a modelação de sistemas híbridos. Uma maneira de utilizar as PN para a modelação de sistemas híbridos é a utilização de DAEs para a parte contínua e uma PN segura (uma só marca) para modelar a parte

discreta [Champagnat98]. Outra das hipóteses consiste em considerar uma configuração da PN como um estado discreto de um HA, que “contém” um DAEs para ser resolvido enquanto a configuração estiver activa. Noutras implementações, em que a PN é usada para a modelação da parte discreta do modelo, é introduzida a noção de habilitar ou inibir a transição à custa de condições extraídas do sistema contínuo [Mosterman98]; neste caso, a presença de uma marca no lugar é comunicada à parte contínua. Anderson, em [Anderson94], usa o método Grafset (PN generalizadas) para modelar a parte discreta de um sistema contínuo.

## **O formalismo dos Statecharts**

O formalismo gráfico conhecido como Statecharts foi proposto inicialmente por Harel [Harel 87] como um método de especificação de sistemas reactivos complexos. Os Statecharts acrescentam aos tradicionais estados e transições do formalismo das Máquinas de Estados Finitos (FSMs) [Hopcroft79] a hierarquia de estados, concorrência ou ortogonalidade entre estados, e um mecanismo de propagação de eventos (*broadcast*) que possibilita a comunicação entre estados. Os mecanismos de hierarquia são especialmente importantes para a modelação do comportamento de sistemas complexos, pois permitem criar estados abstractos de alto nível. Os mecanismos de paralelismo e de propagação de eventos facilitam a modelação do comportamento de sistemas de tempo real. Com estas extensões, Harel refere que é possível a elaboração de diagramas concisos, claros e intuitivos. Sendo um formalismo, os Statecharts possuem sintaxe e semântica próprias que podem ser descritas formalmente permitindo, assim, uma análise e validação automática e formal dos modelos.

São várias as áreas de engenharia onde é possível aplicar o formalismo dos Statecharts, quer para a especificação de controladores, quer para a descrição do comportamento de sistemas. De entre essas áreas destacam-se a indústria aeroespacial [Weber96], comunicações e transportes [Harel97a], [Hedenetz98], os sistemas de tempo real [Beek93] e o hardware e protocolos de comunicação [Durinsky89]. Durante os últimos anos desenvolveram-se várias plataformas de modelação e simulação que incluem o formalismo dos Statecharts. Destacam-se duas aproximações: uso dos Statecharts para a modelação de comportamentos complexos de classes orientadas por objectos – um statechart “ligado” a uma classe especifica todos os aspectos comportamentais dos objectos instanciados dessa classe (exemplos deste tipo de aproximação são as plataformas Objectcharts [Coleman92], ROOM [Selic94], Rhapsody [Harel96], ou UML [Harel97a]); a outra aproximação é a utilização dos Statecharts dentro de plataformas para geração automática de código executável (exemplos desta aproximação são o Stateflow [Stateflow97], Statemate [Harel90], ou BetterState [Drusinsky97]). Normalmente todas estas plataformas oferecem ferramentas para o desenvolvimento, edição, análise e execução de Statecharts.

Desde o seu aparecimento, foram propostas algumas extensões aos Statecharts, como inicialmente definidos por Harel, tais como Statecharts com sobreposição de estados [Harel92], Statecharts temporais e Statecharts híbridos [Kesten92]. Enquanto que a sintaxe dos Statecharts foi concertada muito cedo, a definição da semântica formal não está ainda estabelecida. De facto os Statecharts não têm uma semântica oficial [Harel96] e, por isso, dependendo do tipo de sistemas a especificar ou modelar, foram propostas algumas variantes dos Statecharts. Beeck, em [Beeck94], apresenta uma comparação entre variantes de Statecharts. Especificações do formalismo dos Statecharts, com exemplos variados da sua utilização, podem ser encontradas nos trabalhos realizados em [Harel87], [Harel96], [Harel97b] ou [Huizing91].

## 2.6 A linguagem Modelica

Nesta secção, os conceitos da linguagem Modelica são apresentados recorrendo a exemplos que evidenciam as características da linguagem. A maioria dos exemplos visam a modelação de componentes de sistemas hidráulicos onde se recorre, sempre que possível, ao exemplo do servomecanismo hidráulico apresentado na secção 2.3.3. O tutorial da linguagem [ModTutorial00], versão 1.4 foi usado como base. Para uma análise mais detalhada propõe-se a consulta ao documento da especificação da linguagem [ModSpecific00].

### Elementos básicos da linguagem

O suporte de modelos complexos implica uma estruturação cuidadosa dos mesmos de modo a permitir a sua reutilização. Um modelo pode então ser constituído por componentes básicos como *Real*, *Integer* ou *String*, por componentes estruturais para a criação de estruturas hierárquicas, matrizes ou listas de modelos, equações e/ou algoritmos, ligações entre modelos e funções. A declaração de propriedades de variáveis também é necessária, pois existem diferentes tipos de variáveis. As variáveis reais têm um atributo, chamado *start* para estabelecer um valor inicial da variável. *Real* é um tipo de dados pré definido, (no entanto é possível definir novos tipos) onde se pode inserir informação útil como são as unidades e o tipo de grandeza da nova variável.

Novos tipos de dados para representar pressões e caudais podem ser definidos da seguinte forma:

```
type Pressure = Real (quantity = "Pressure", unit = "Pa");
```

```
type VolumeFlowRate = Real (quantity = "VolumeFlowRate", unit = "m3/s");
```



Assim, a declaração de uma variável para representar a pressão na câmara de um cilindro poderia ser declarada e inicializada por:

```
Pressure P1 (start = 35e5);
```

Uma declaração pode ser precedida pelas palavras reservadas **constant** ou **parameter** para indicar que o componente declarado é constante.

```
parameter Pressure Ps = 70e5;
```

As equações são compostas por expressões em ambos os lados do sinal de igualdade:

```
T*der(u) + y = u;
```

### Classes para re-utilização dos modelos

Os volumes de óleo compressível são usuais na modelação de sistemas hidráulicos através da utilização de técnicas de modelação com parâmetros concentrados. É assim mais conveniente definir um volume de óleo constante e criar as instâncias necessárias no circuito hidráulico em vez de repetir a equação de continuidade [Merritt67]. Esta definição pode ser realizada através de uma classe. A declaração de uma classe contém uma lista de declarações de componentes e uma lista de equações precedidas pela palavra-chave *equation*.

```
class OilVolume
  parameter Real vol=0.5e-4 “Volume à pressão atmosférica”;
  parameter Real beta = 1e9 “Módulo de compressibilidade”;
  Pressure p;
  VolumeFlowRate q;
equation
  der(p) = beta/vol * q;
end OilVolume;
```

A classe pode então ser usada para criar novas instâncias com diferentes volumes e/ou módulo de compressibilidade.

```
OilVolume V1(vol = 0.8e-4), V2(vol = 6e-4, beta = 1.2e9);
```

Neste caso, foi usada uma modificação para alterar os volumes e o módulo de compressibilidade para valores diferentes dos valores atribuídos por defeito.

O conceito de classe é semelhante ao usado em linguagens de programação orientadas por objectos. Este conceito é usado para vários propósitos, nomeadamente para modelar componentes,

implementar mecanismos de ligações, agrupar conjuntos de parâmetros ou para implementação de blocos entrada/saída ou funções. São fornecidas palavras-chave especiais para a representação de classes com uso específico: *model*, *connector*, *record*, *block*, *function*, *type* e *package*. Algumas destas classes são especiais; por exemplo, um *record* não pode conter equações e uma *package* só pode conter declarações de classes e constantes.

Palavra chave	Regras
model	Não pode ser usado em ligações.
connector	Não são possíveis equações em nenhum dos seus componentes.
record	Não pode conter equações; não pode ser usado em ligações.
block	Bloco de entrada/saída com causalidade fixa. Cada componente da interface também tem de ter causalidade.
package	Só pode ter declarações de classes e constantes.
function	Tem as mesmas restrições do <i>block</i> ; não pode ter equações, somente algoritmos.

### Ligações entre modelos

As classes permitem a construção hierárquica de modelos. Contudo, também é possível definir ligações físicas entre modelos através de uma classe restrita, *connector*. Esta funcionalidade permite a construção de modelos à custa de componentes mais simples interligados. Nos sistemas hidráulicos, as ligações são caracterizadas por duas variáveis físicas, pressão e caudal volumétrico, as quais devem ser tratadas de forma diferente. De facto, enquanto as pressões deverão ser iguais, os caudais volumétricos, supondo incompressibilidade do fluido, obedecem à lei de conservação das massas e, nesse sentido, deverão ser somados e igualadas a zero. Este tipo de variáveis deverá ser declarado com o prefixo *flow*.

```

connector HydConnector
    Pressure p;
    flow VolumeFlowRate q;
end HydConnector;

```

desta forma, a classe OilVolume poderia ser descrita usando uma ligação hidráulica

```

class OilVolume
  HydConnector HydA;
  parameter Real vol=0.5e-4 "Volume à pressão atmosférica";
  parameter Real beta = 1e9 "Módulo de compressibilidade";
  Pressure p;
  VolumeFlowRate q;
equation
  HydA.p = p;
  HydA.q = q;
  der(p) = beta/vol * q;
end OilVolume;

```

e uma resistência hidráulica com escoamento laminar e volumes concentrados nas suas ligações poderia ser descrita pelo modelo seguinte:

**type** Conductance = Real (**quantity**="Conductance", **unit**="m<sup>3</sup>/(s\*Pa)");

```

model ResLaminar
  HydConnector inRes, outRes;
  parameter Conductance G = 4.2e-13;
  OilVolume VolIn(vol = 1e-6);
  OilVolume VolOut(vol = 1e-6);
  Pressure dp "diferença de pressão";
  VolumeFlowRate q "caudal volumétrico através do elemento ";
equation
  connect(VolIn.port, inRes);
  connect(VolOut.port, outRes);
  dp = inRes.p - outRes.p;
  q = G*dp;
  q = inRes.q;
  inRes.q + outRes.q = 0;
end ResLaminar;

```

*connect* é um operador especial que gera equações tendo em conta os tipos de variáveis que estão envolvidas. Neste caso, *connect*(VolIn.port, inRes) gera as equações:

$$\text{VolIn.port.p} = \text{inRes.p};$$

$$\text{VolIn.port.q} + \text{inRes.q} = 0;$$

## Modelos parciais e herança

Uma característica importante, que permite construir descrições reutilizáveis, é a definição e reutilização de modelos parciais. Como existem vários componentes hidráulicos com duas ligações, como uma válvula de segurança ou uma linha de transmissão hidráulica, pode-se definir um componente genérico com duas ligações hidráulicas como sendo uma base para alguns componentes com duas ligações. Assim, para componentes em que se assume que não existe variação interna de volume (caudais de entrada e de saída iguais), o modelo poderia ser o seguinte:

```
model TwoPortHydComp
  parameter Real rho = 865 “ massa específica do óleo (kg/m3)”;
  HydConnector HydA;
  HydConnector HydB;
  Pressure dp "diferença de pressão ";
  VolumeFlowRate q "caudal volumétrico através do componente ";
equation
  dp = HydA.p - HydB.p;
  q = HydB.q;
  0 = HydA.q + HydB.q;
end TwoPortHydComp;
```

Desta forma, os componentes hidráulicos com dois terminais de ligação poderiam herdar as características comuns definidas na classe base de componentes com dois terminais. Por exemplo, um orifício com escoamento turbulento poderia ser modelado pela classe:

```
model TurbulentOrifice
  extends TwoPortHydComp;
  parameter Real Cd = 0.7 “ coeficiente de descarga do orifício”;
  parameter Real A0 = 1e-5 “secção de passagem”;
equation
  q = A0*Cd*sqrt(dp/rho);
end TurbulentOrifice;
```

Esta facilidade permite implementar o conceito de herança das linguagens orientadas por objectos. Neste caso, a múltipla herança pode ser implementada através de várias instruções *extends*. No exemplo anterior, a classe *TurbulentOrifice* herda todas as características, parâmetros, variáveis e

equações da classe *TwoPortHydComp* acrescentando-lhe as equações, variáveis e parâmetros necessários à modelação de orifícios com escoamento turbulento.

## Parametrização de classes

Em Modelica, o conceito de parametrização não envolve somente constantes e matrizes, mas também classes. Considere-se um circuito hidráulico muito simples composto por uma bomba de pressão constante, um orifício e um tanque:

```
model CtePressurePump
  parameter Pressure ctePressure=100e5;
  HydConnector HydA;
  Pressure p;
  VolumeFlowRate q;
equation
  p = HydA.p;
  q = HydA.q;
  p = ctePressure;
end CtePressurePump;
```

```
model HydTank
  HydConnector HydA;
equation
  HydA.p = 0;
end HydTank;
```

```
model hydCircuit
  CtePressurePump Pump;
  HydTank Tank;
  replaceable TurbulentOrifice tOri(Cd = 0.7);
equation
  connect(Pump.HydA,tOri.HydA);
  connect(tOri.HydB,Tank.HydA);
end hydCircuit;
```

Suponha-se, agora, que se pretende usar a topologia do circuito hidráulico, mas mudando o componente orifício por um idêntico mas com secção variável:

```

model TurbulentOrificeVarSection
  extends TwoPortHydComp;
  parameter Real Cd = 0.7;
  parameter Real A0 = 1e-5;
  Real percentClose;
equation
  q =(A0 - percentClose * A0)*Cd*sqrt(dp);
end TurbulentOrificeVarSection;

```

Isto é possível através da re-declaração do orifício *tOri*, uma vez que as classes *TurbulentOrificeVarSection* e *TurbulentOrifice* são ambas um sub-tipo de *TwoPortHydComp*.

```

model hydCircuitExt
  extends hydCircuit(redeclare TurbulentOrificeVarSection tOri(A0 = 2e-5) );
equation
  tOri.percentClose = sin (2*3.14 * time);
end hydCircuitExt;

```

Esta é uma mudança significativa no modelo do circuito, havendo mesmo a possibilidade de o tornar inválido. Por essa razão, o modelador tem de permitir essa modificação declarando o componente como substituível:

```
replaceable TurbulentOrifice tOri(Cd = 0.7);
```

e o utilizador do circuito tem de marcar o componente através da sigla **redeclare**.

## Matrizes e vectores

Uma variável do tipo array pode ser declarada de duas formas diferentes:

```
Real [3] position, velocity, acceleration; Real [3,2,8] table;
```

ou então

```
Real position[3], velocity [3], acceleration[3], table[3,2,8];
```

É também possível definir um novo tipo para uma matriz, útil para a modelação da dinâmica de elementos 3D:

```
type Position = Real(unit="m");
```

```
type Position3D = Position[3];
```

## Diagramas de blocos

O conceito de classe pode ser usado para modelar diagramas de blocos como um caso especial. É então possível definir a direcção do fluxo de dados através dos prefixos *input* e *output* nas declarações. Isto também permite averiguar se só uma ligação é feita a uma entrada, se as saídas não estão ligadas a saídas e as entradas não estão ligadas a entradas.

Um sistema representado em espaço de estados, (2.24), sendo um bloco com entradas e saídas, pode ser descrito da seguinte forma:

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx + Du \end{aligned} \tag{2.24}$$

```
block StateSpace "espaço de estados linear"
  parameter Real A[ : , : ] "Matriz A do modelo de espaço de estados";
  parameter Real B[size(A, 1), : ] "Matriz B";
  parameter Real C[ : , size(A, 2) ] "Matriz C";
  parameter Real D[size(C, 1), size(B, 2)]=zeros(size(C, 1), size(B, 2)) "Matriz D";
  input Real u[size(B, 2)]; "vector de entrada";
  output Real x[size(C, 1)]; "vector de estado";
protected
  output Real y[size(A, 1)] "vector de saída";
equation
  assert (size(A,1) == size(A,2), "A matriz A tem de ser quadrada");
  der(x) = A*x + B*u;
  y = C*x + D*u;
end StateSpace;
```

Uma matriz pode ser declarada sem dimensões específicas, substituindo as suas dimensões por A[:,:]. As dimensões actuais de uma matriz podem ser devolvidas pela função *size*.

*Assert* é uma função pré-definida para fornecer mensagens de erro. Num modelo, a informação também pode ser protegida dos utilizadores através do cabeçalho *protected*. As dimensões de A, B e C são dadas implicitamente pelos parâmetros das matrizes usados na instanciação do modelo *StateSpace*. A matriz D, por defeito, é uma matriz de zeros.

O bloco seguinte poderia ser usado para testar o bloco representativo do espaço de estados:

```

block TestSpaceSpace
    StateSpace S(A = [0.1, 2; 3, 1.5], B=[2, 7; 3, 1], C=[0.1,2]);
equation
    S.u = { time, sin(time) };
end TestSpaceSpace;

```

A variável *time* é a variável independente e está disponível em todas as classes sem necessidade de declaração. A classe do tipo *block* foi introduzida para permitir um melhor diagnóstico em modelos de componentes de entradas/saídas puros. Neste caso, a correção do componente pode ser analisada localmente, o que não é possível para componentes onde a causalidade das variáveis é desconhecida.

### Estruturas de repetição, algoritmos e funções

É também possível indexar expressões através da estrutura repetitiva *for*. O exemplo seguinte usa um ciclo repetitivo para gerar várias equações que avaliam as potências de *x*. Esta estrutura pode então ser usada num bloco para avaliação polinomial.

```

block PolinomialEvaluation
    parameter Real a[:];
    input Real x;
    output Real y;
protected
    parameter Integer n = size(a,1) -1;
    Real xPowers[n+1];
equation
    xPowers[1] = 1;
    for i in 1:n loop
        xPowers[i+1] = xPowers[i]*x;
    end for;
    y = a * xPowers;
end PolinomialEvaluation;

```

Desta forma, o bloco poderia ser usado para cálculo da exponencial de *t* usando os primeiros cinco termos da expansão em série de Taylor.



```

model Exponential
  PolynomialEvaluation exp_x(a={ 1, 1, 1/2, 1/6, 1/24});
  Real res;
equation
  exp_x.x = time;
  res = exp_x.y;
end Exponential;

```

O mecanismo básico de descrição em Modelica são equações e não instruções de atribuição. Isto oferece a flexibilidade necessária para, por exemplo, a descrição de um componente poder ser usada com diferentes causalidades dependendo da forma como é ligado. Contudo, em algumas situações é mais conveniente usar instruções de atribuição; por exemplo, seria mais natural definir um controlador digital com um conjunto de instruções ordenadas já que, na prática, ele vai ser implementado dessa forma. A linguagem Modelica permite a chamada de funções escritas noutras linguagens, mas a dificuldade na detecção de erros de programação pode levar a falhas na simulação. Na maior parte dos casos a solução passa pela utilização de algoritmos em Modelica pois, para programadores inexperientes são mais seguros do que a chamada a funções externas.

O mecanismo recursivo usado no exemplo anterior é mais eficiente quando expresso por um algoritmo, isto é, uma sequência de instruções de atribuição, instruções condicionais ou repetitivas.

```

algorithm
  y := 0;
  xPower := 1;
  for i in 1:n loop
    y := y + a[i]*xPower;
    xPower := xPower*x;
  end for;

```

Um algoritmo em Modelica é uma função, sem memória interna e efeitos laterais, isto é, sempre que um algoritmo é usado com as mesmas entradas, o resultado será o mesmo; se uma função for chamada durante o processo de integração contínuo isso é um pré-requisito absoluto. A secção de algoritmos num modelo é definida pela palavra-chave *algorithm*. Para a distinção entre as instruções de atribuição e as equações são usados os operadores atribuição “:=” e identidade “=” respectivamente.

O exemplo anterior é um bloco de entradas/saídas especial pois não contém nenhum estado. Já que o bloco não tem memória interna poder-se-á invocar o bloco com uma função, isto é, a memória

para as variáveis é alocada temporariamente enquanto o algoritmo da função está a ser executado. A linguagem Modelica permite uma especialização de uma classe chamada *function* a qual tem somente entradas, saídas, um algoritmo e nenhuma equações.

```
function PolinomialEvaluation2
  parameter Real a[:];
  input Real x;
  output Real y;
protected
  parameter Integer n = size(a,1) -1;
  Real xPowers[n+1];
algorithm
  y := 0;
  xPower := 1;
  for i in 1:n loop
    y := y + a[i]*xPower;
    xPower := xPower*x;
  end for;
end PolinomialEvaluation2;
```

A declaração de uma função é semelhante à declaração de uma classe, mas inicia-se com a palavra reservada **function**. Os argumentos de entrada são definidos pela palavra reservada **input** e os argumentos de saídas são definidos pela palavra **output**. Em vez de criar uma instância, como acontecia com um bloco, é possível invocar a função da forma tradicional com a associação dos argumentos de entrada com a sua posição na chamada da função:

```
Exp_x = PolinomialEvaluation2 ({1, 1, 1/2, 1/6, 1/24},time);
```

Como já foi referido, é também possível chamar funções definidas no exterior da linguagem Modelica. O corpo de uma função externa é definido pela palavra-chave **external**. A linguagem C é a linguagem por defeito para a invocação de funções externas, no entanto, também podem ser invocadas funções em FORTRAN:

```
function log
  input Real x;
  output Real y;
  external "FORTRAN 77";
end log;
```

## Modelos híbridos

A linguagem Modelica pode ser usada para modelos contínuos e/ou discretos. Para a parte discreta é usado o princípio de fluxo de dados síncrono com a regra de atribuição simples. Isto enquadra-se nos sistemas de equações diferenciais e algébricas com o mesmo número de equações e incógnitas. Este tipo de implementação segue os princípios de linguagens síncronas [Halbwachs 93], como a linguagem Signal [Benveniste90], Lustre [Halbwachs91] ou Esterel [Berry92].

## Modelos descontínuos

As expressões *if-then-else* permitem a descrição de fenómenos com expressões diferentes em diferentes regiões de operação, isto é, permitem sistemas de equações com estrutura variável. Por exemplo, uma válvula limitadora de pressão pode ser modelada de uma forma estática [Beater98], através da utilização de expressões diferentes se a válvula estiver aberta ou fechada.

```
model ReliefValve "modelo de uma válvula limitadora de pressão"  
extends TwoPortHydComp;  
parameter Pressure pClose(min=0) = 70e5 "válvula fechada se dp for menor ";  
parameter Pressure pOpen = 75e5 "válvula totalmente aberta se dp for maior ";  
parameter Real gLeak = 1e-12 "condutância de fugas quando a válvula está fechada ";  
parameter Real gOpen=1.7e-9 "condutância se a válvula está totalmente aberta";  
Boolean close(start=false) "indica se a válvula está fechada";  
Boolean open(start=false) " indica se a válvula está totalmente aberta";  
equation  
  close = dp < pClose;  
  open = dp > pOpen;  
  q = if close then dp*gLeak  
    else (dp - pClose)*gOpen + dp*gLeak;  
end ReliefValve;
```

Esta construção permite a introdução de descontinuidades nos modelos, bem como mudanças mais drásticas tais como a substituição de conjuntos de equações por outros dependendo de uma determinada condição.

## Modelos condicionais

A construção de modelos com diferentes complexidades pode ser efectuada através de componentes condicionais em Modelica, como se mostra no próximo modelo onde os controladores, com complexidades diferentes, são modelados como sub-componentes:

```
block HydController
  input Boolean simple = true;
  input Real error;
  output Real y;
protected
  ControllerP c1(u = error, enable = simple, reset=true);
  ControllerPID c2(u = error, enable = not simple, reset=true);
equation
  y = if simple then c1.y else c2.y;
end HydController;
```

O atributo *enable* é uma entrada do tipo Boolean inerente a qualquer bloco com a equação “*enable=true*” definida por defeito. A condição de *enable* pode ser dependente do tempo ou do estado actual do modelo. Se “*enable = false*”, para uma instância de um modelo, as suas equações não são avaliadas, todas as variáveis declaradas são consideradas constantes e todos os seus sub-componentes são inibidos. O atributo de *reset* é propagado a todos os sub-componentes, e permite inicializar todas as variáveis antes de habilitar o modelo.

**Modelos discretos no tempo e de eventos discretos**

As acções a serem realizadas na ocorrência de eventos são especificadas por instruções **when**:

```
when condition then
  equations;
end when;
```

As equações são activadas instantaneamente quando a condição tem o valor lógico verdadeiro. Podem ser realizadas acções especiais quando a simulação se inicia e quando termina através do teste das funções *initial()* e *terminal()*. Um operador especial pode ser usado para atribuir, num evento, novos valores a estados contínuos do modelo.

Consideremos os sistemas discretos no tempo ou os sistemas amostrados. Estes são caracterizados pela possibilidade de, periodicamente, amostrar variáveis contínuas de entrada, calcular novas saídas que influenciam a parte contínua do modelo e actualizar as variáveis de estado discretas. As variáveis de saída mantêm os seus valores entre amostragens. É então necessário activar as equações discretas uma vez em cada amostragem. Esta activação pode ser executada de uma função pré definida, **sample**(*start*, *period*), que activa a condição da instrução **when** quando  $time = start + n \cdot period$  com  $n \geq 0$ . Desta forma, um modelo discreto de primeira ordem, descrito pelo seu espaço de estados, pode ser escrito como:

```

block DiscreteStateSpace
  parameter Real a, b, c, d, Period=1;
  input Real u;
  discrete output Real y;
protected
  discrete Real x;
equation
  when sample(0, Period) then
    x = a*pre(x) + b*u;
    y = c*pre(x) + d*u;
  end when;
end DiscreteStateSpace;

```

A notação especial **pre(x)** é usada para representar o valor da variável de estado discreta  $x$  imediatamente antes da ocorrência do evento. Neste caso, a primeira amostragem é efectuada quando a simulação se inicia, pois, com  $start > 0$  não existe nenhuma equação para definir  $x$  e  $y$  inicialmente. Todas as variáveis definidas por instruções **when** mantêm os seus valores entre activações das equações e, por serem discretas, devem ter o prefixo **discrete**.

Para amostragens não periódicas é forçoso utilizar um outro método para especificar as amostragens. A sequência de instantes de amostragem pode ser calculada pelo próprio modelo e guardada numa variável discreta, por exemplo *NextSampling*. Um conjunto de equações poderia então ser activado quando a condição  $time \geq NextSampling$  fosse verdadeira. Uma formulação alternativa para o sistema discreto anteriormente referenciado seria:

```

block DiscreteStateSpace2
  parameter Real a, b, c, d, Period=1;
  input Real u;
  discrete output Real y;
protected
  discrete Real x, NextSampling(start=0);
equation
  when time >= pre(NextSampling) then
    x = a*pre(x) + b*u;  y = c*pre(x) + d*u;
    NextSampling = time + Period;
  end when;
end DiscreteStateSpace2;

```

## Modelos de componentes ideais (válvula de esfera)

Uma válvula de retenção pode ser modelada como um componente ideal. O caudal através da válvula,  $q$ , não pode ser definido em função de  $dp$  (ou vice versa) devido às características ideais da válvula. Contudo, para estas curvas de comportamento planares, pode ser usada uma forma paramétrica com:

$q = f\langle s \rangle$   
 $dp = g\langle s \rangle$  onde  $s$  é um parâmetro escalar da curva da figura seguinte:

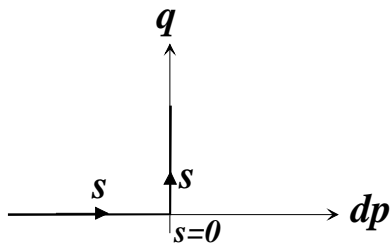


Figura 2.9 Curva planar que descreve o comportamento de uma válvula de retenção ideal

O modelo completo da válvula de esfera ideal seria:

```
model ShuttleValve "modelo ideal de uma válvula de retenção"  
  extends TwoPortHydComp;  
protected  
  Real s;  
equation  
  q = if s>0 then s else 0;  
  dp = if s>0 then 0 else s;  
end ShuttleValve;
```



## **3 METODOLOGIA DE MODELAÇÃO: STATECHARTS E MODELICA**

Neste capítulo apresenta-se a metodologia de modelação proposta para o desenvolvimento e gestão de bibliotecas de modelos para componentes de sistemas hidráulicos. São também apresentadas as ferramentas desenvolvidas para a implementação da metodologia. Nesse contexto, é apresentada a biblioteca de modelos desenvolvida para a implementação do formalismo dos Statecharts na linguagem Modelica.

### **3.1 Statecharts**

#### **3.1.1 Considerações informais sobre a sintaxe e a semântica dos Statecharts**

Nesta secção apresentam-se as características essenciais dos Statecharts, muito especialmente dos Statecharts híbridos, ao nível sintáctico e semântico, com vista à sua implementação em linguagem Modelica.

Um statechart é descrito através de estados, transições, eventos e condições. Os estados e eventos são considerados a maneira mais natural de descrição da dinâmica de sistemas complexos. Contudo, em sistemas complexos poderão existir dezenas de estados distintos que, organizados de acordo com o formalismo de um só nível como é o formalismo das FSMs, resultam em diagramas de estado complicados e difíceis de analisar; de facto, pode ser difícil, senão mesmo impossível, mostrar todos os estados num único diagrama. A dependência e o paralelismo de acções de subsistemas concorrentes aumenta exponencialmente o número de estados e transições no modelo FSMs. O mesmo comportamento da FSM da Figura 3.1 pode ser descrito num statechart com metade das transições, já que os Statecharts oferecem construções próprias para representar subsistemas concorrentes.



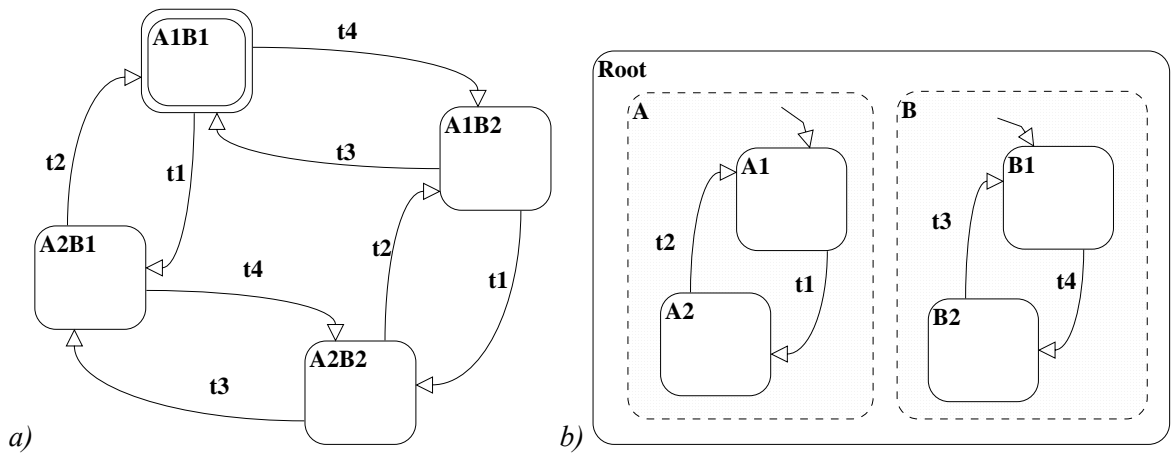


Figura 3.1 Descrição de actividades paralelas com a) FSM, b) Statecharts

Os estados permitem a especificação de diferentes contextos do sistema e são representados graficamente através de rectângulos. Quando um estado está activo descreve uma determinada situação no sistema representado pelo statechart. Existem três tipos de estados: básicos, OR e AND. No exemplo da Figura 3.2 os estados “A1”, “A2”, “A3”, “B11”, “B12”, “B21”, “B22”, “B23”, “B311”, “B312” e “B32” são estados básicos.

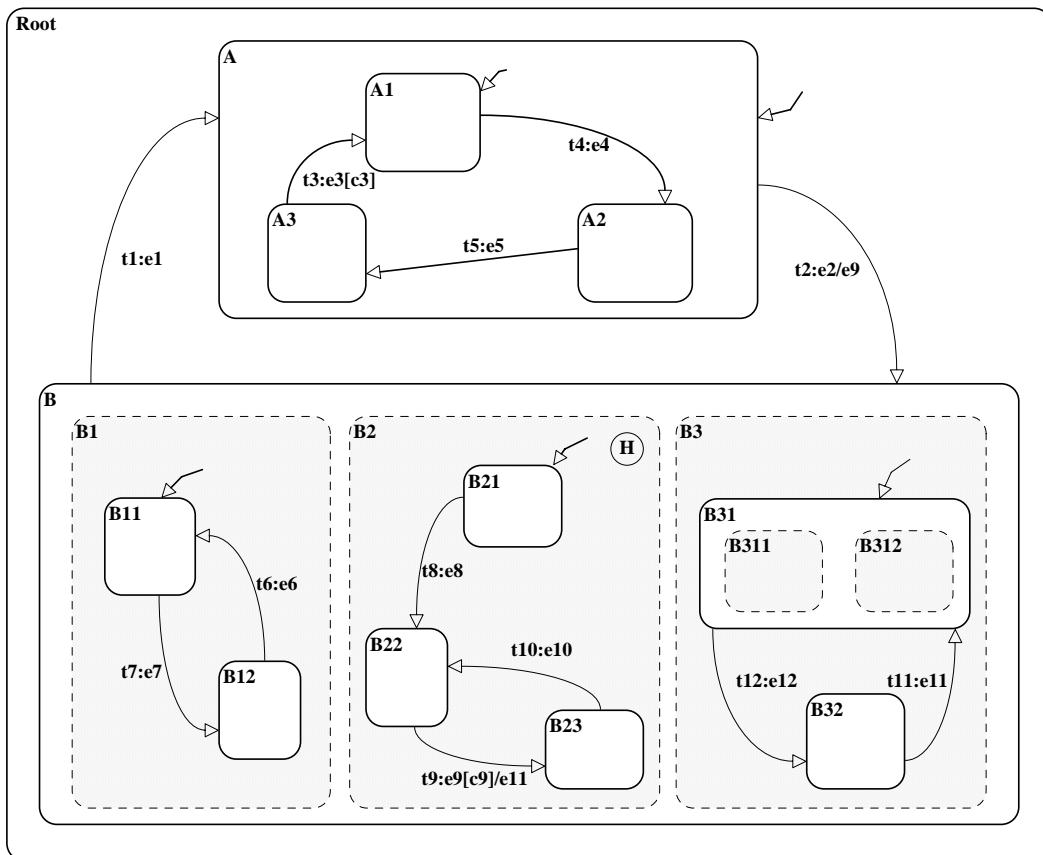


Figura 3.2 Exemplo de um statechart com diferentes níveis hierárquicos

Uma das diferenças, relativamente às FSMs, é a possibilidade de decomposição hierárquica de estados, permitindo assim o agrupamento de estados noutros estados formando diferentes níveis hierárquicos. Esta forma hierárquica de organizar informação encoraja a utilização de ferramentas de ampliação para a movimentação fácil entre níveis de abstracção. O formalismo dos Statecharts introduz o conceito de profundidade nos diagramas de estados através da decomposição OR de um estado FSM convencional. A ortogonalidade representa o dual da profundidade e corresponde a uma decomposição AND de um FSM.

A decomposição OR ou AND de estados dá origem a sub-estados. Um estado é um sub-estado de outro quando este for ancestral directo ou indirecto ao longo da hierarquia. Os estados "Root", "A", "B1", "B2" e "B3" são decomposições do tipo OR. O estado "B1" contém os sub-estados "B11" e "B12", que são denominados estados filhos de "B1", sendo este o seu estado pai. Um statechart possui sempre um estado sem estado pai (neste caso o estado "Root") a que se chama raiz do statechart. Quando um estado OR está activo um e só um sub-estado está também activo. Tomando o exemplo, quando "A" está activo, somente um dos estados "A1", "A2" ou "A3" está activo. A concorrência dentro de um statechart pode ser descrita através de estados AND, permitindo a descrição de actividades concorrentes num mesmo modelo através de estados ortogonais. Estes estados ortogonais são todos simultaneamente activados, ou desactivados, aquando da entrada, ou saída, de um estado AND. A este tipo de activação e desactivação dos sub-estados chama-se activação e desactivação hierárquica. Os estados "B1", "B2" e "B3" são sub-estados ortogonais resultantes da decomposição AND do estado "B". O processo de decomposição AND e OR de estados pode prosseguir até que seja atingida a descrição de baixo nível pretendida. Quando a modelação de sistemas produz statecharts complexos, pode-se decompor o statechart em diagramas separados conservando-se a mesma hierarquia. Esses diagramas podem ser visualizados através de ferramentas de ampliação, normalmente fornecidas pelos editores gráficos.

As transições são consideradas o único meio pelo qual um estado pode ser activado explicita ou implicitamente. Desta forma, a interacção e a evolução de um statechart são geridas por eventos internos ou externos que originam o disparo de transições e, subsequentemente, o relacionamento sequencial entre estados. Os eventos podem também estar associados às próprias transições através da execução de acções associadas ao seu disparo, como são o caso das transições "t9" e "t2" que têm associadas as acções "e11" e "e9" respectivamente. As transições são representadas por setas unidireccionais que ligam os estados origem aos estados destino. Algumas implementações (Statemate, Stateflow) do formalismo dos Statecharts permitem a utilização de transições compostas (combinação de segmentos de transições de modo a formar transições completas). Na presente proposta não são implementadas as transições compostas, considerando-se que, neste caso só têm um estado origem e um estado destino. Existem, porém, as transições denominadas *default*,

que só têm um estado destino e que são disparadas quando um estado OR é activado. Por exemplo, quando o estado “B1” é activado a transição *default* é disparada activando o seu estado destino, “B11”, que é considerado o estado *default* de “B”. A sintaxe genérica de rótulo de uma transição num statechart é “*event[guard]/action*”, onde *event* é uma expressão contendo os eventos necessários para que a transição ocorra, *guard* é uma expressão condicional que protege a transição inibindo-a do disparo caso o seu valor lógico seja falso. A expressão *guard* pode ser uma especificação de um domínio de uma variável, a avaliação dos componentes de um estado ortogonal ou pode ser usada para estabelecer prioridades nas transições. Em resumo, uma transição só pode ser disparada quando o seu estado origem estiver activo, ocorram os eventos definidos em *event* e a expressão condicional *guard* seja verdadeira. O disparo duma transição provoca a desactivação do seu estado origem e a activação do seu estado destino. Aquando do disparo duma transição, as acções definidas na expressão *action* são executadas. Estes mecanismos de geração de eventos, associados ao disparo de transições, permitem especificar comunicações entre componentes ortogonais e podem ser usados para a sincronização de tarefas concorrentes. Outro tipo de eventos é despoletado por acções externas, por exemplo, no caso de modelos de sistemas reactivos, devido a acções dos utilizadores da aplicação.

### 3.1.2 Conflito em transições e não determinismo

O não determinismo tem lugar quando duas ou mais transições com o mesmo estado origem podem ser disparadas no mesmo instante. Duas ou mais transições estão em conflito quando estão preparadas para o disparo simultâneo, com possibilidade de desactivar o mesmo estado origem e activar dois sub-estados destino diferentes de um mesmo estado OR. Por exemplo, na Figura 3.3 as transições “t1” e “t2” estão em conflito pois implicam a desactivação do mesmo estado “A1”.

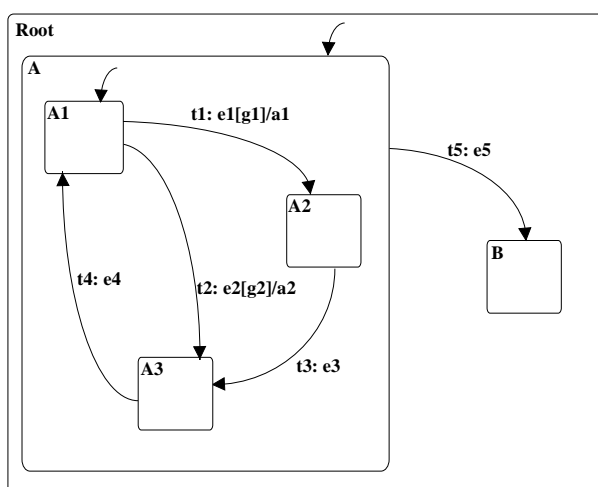


Figura 3.3 Conflito de transições e não-determinismo

A transição “t5” também está em conflito com todas as outras transições pois uma das condições para “t5” ser disparada é o estado A estar activo e, portanto, também um dos seus sub-estados está activo (o que habilita outras transições).

Considerando as transições “t1” e “t2”, em que ambos os eventos “e1” e “e2” são verdadeiros, verifica-se um caso de não determinismo já que não existe razão para preferir uma transição em relação à outra. Por outro lado, se as duas transições fossem disparadas isto conduzia a uma situação em que dois sub-estados estariam activos num estado OR, o que não é permitido no formalismo dos Statecharts. Algumas situações de não determinismo podem ser eliminadas dum statechart através da atribuição de prioridades às transições. Por exemplo, se se pretendesse dar prioridade ao disparo de “t1” relativamente a “t2” poder-se-ia estabelecer essa prioridade na condição de protecção da transição “t2”, definindo o seu rótulo da seguinte forma: “t2: e2[g2 **and not** e1]”.

Outro caso de conflito entre transições ocorre devido à própria estrutura do statechart. No presente exemplo, “t5” pode estar habilitada a disparar ao mesmo tempo que qualquer uma das outras transições. Pode-se então usar o conceito de domínio (*scope*) de uma transição para definir as prioridades dum transições em relação às outras; define-se domínio de uma transição como o estado OR, ao nível mais baixo na hierarquia, que é o ancestral comum aos estados destino e origem da transição. Neste caso, o domínio de “t5” é o estado “Root”, enquanto que o domínio de “t1”, “t2”, “t3” e “t4” é o estado “A1”. É comum estabelecer prioridades mais elevadas para as transições cujo domínio corresponde a estados OR com níveis mais altos na hierarquia; este tipo de prioridades também é conhecido como prioridade estrutural.

### 3.1.3 O conceito de história

Um dos conceitos adicionais do formalismo dos Statecharts é o conceito de história que permite um modo diferente de activação de sub-estados quando um estado OR é activado. Por defeito, o sub-estado a activar, quando um estado OR é activado, é o estado *default* (o estado destino da transição *default*). No entanto, quando o estado OR tem história, representada pela letra “H” inserida num pequeno círculo no interior do estado OR, o sub-estado a ser activado é aquele que esteve activo imediatamente antes da última desactivação hierárquica (desactivação devida à desactivação do estado OR), isto é, o sub-estado mais recentemente “visitado”. Caso seja a primeira activação do estado OR, o seu sub-estado activado será sempre o sub-estado *default*. É possível definir dois tipos de história: história simples (H) e história recursiva (H\*). No mecanismo de história simples considera-se o sub-estado mais recentemente visitado no nível imediatamente inferior ao nível do

estado OR que possui história (H). No caso da história recursiva (H\*) considera-se o estado mais recentemente visitado em todos os níveis hierarquicamente inferiores. Como facilmente se depreende, a história recursiva corresponde à atribuição de história simples a todos os estados OR que tenham como ancestral (directo ou indirecto) o estado OR com história recursiva. No caso da Figura 3.2 o estado OR “B1” tem história simples, portanto quando o estado “B1” é activado, e caso não seja a primeira activação, o sub-estado mais recentemente visitado (“B11” ou “B12”) será aquele que será activado.

### 3.1.4 Representação da estrutura de um statechart

A estrutura geral de um statechart pode ser representada por uma árvore em que as folhas representam os estados básicos [Turine98]. Os nós da árvore representam estados não básicos, AND ou OR, sendo o conceito de profundidade dado pelos diferentes níveis hierárquicos. Para distinção dos estados correspondentes aos nós da árvore, os estados OR são representados pela sigla “v”, enquanto os estados AND são representados pela sigla “^”. Todos os nós e folhas da árvore contêm os nomes dos estados correspondentes. Na Figura 3.4 mostra-se a árvore AND/OR do statechart apresentado na figura Figura 3.2. Em algumas implementações é permitido que o estado origem e o estado destino de uma transição residam em qualquer nível da árvore. Em termos gráficos isto corresponde à possibilidade das transições cruzarem o bordo dos estados. Este tipo de implementação torna o formalismo mais “potente” em termos de especificação de comportamentos, dificultando porém a modularidade dos statecharts e a sua própria implementação [Beeck94].

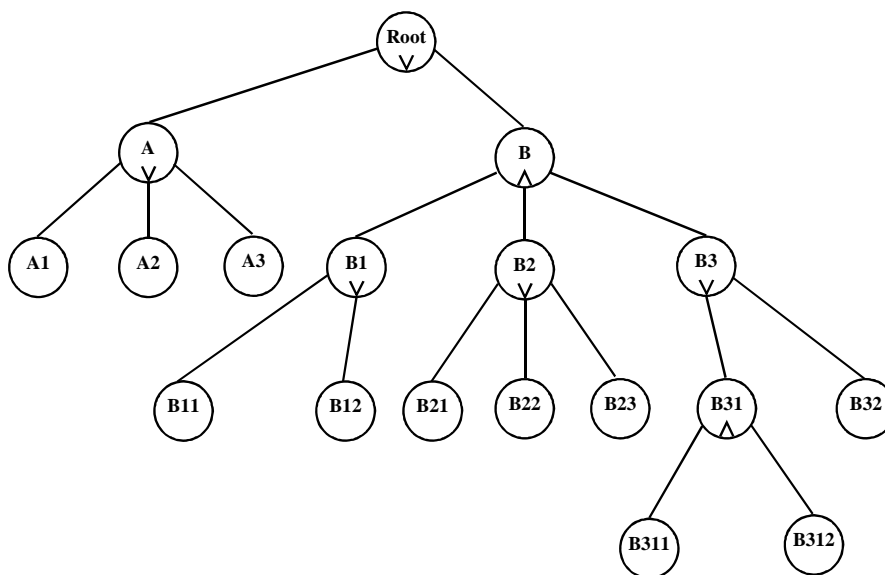


Figura 3.4 Árvore AND/OR do Statechart da Figura 3.2

### 3.1.5 Acções e actividades para modelação de comportamentos híbridos

Relativamente ao FSM, o formalismo dos Statecharts adiciona a possibilidade de associação de acções discretas à activação e à desactivação de estados e ao disparo de transições. Estas acções são executadas quando os respectivos eventos ocorrerem. A associação de actividades contínuas aos estados do statechart permite a especificação de comportamento contínuo quando o respectivo estado está activo. Esta extensão do formalismo dos Statecharts é conhecida como Statecharts Híbridos [Kesten92]. Desta forma, os Statecharts Híbridos adequam-se à modelação de sistemas híbridos: enquanto as acções e transições capturam as características discretas do sistema, as actividades descrevem a sua parte contínua. As actividades contínuas associadas a um estado estão operacionais enquanto o estado estiver activo. Todas as actividades têm um comportamento contínuo no tempo, permitindo então a modelação de sistemas híbridos, como se pode ver no exemplo apresentado na Figura 3.5.

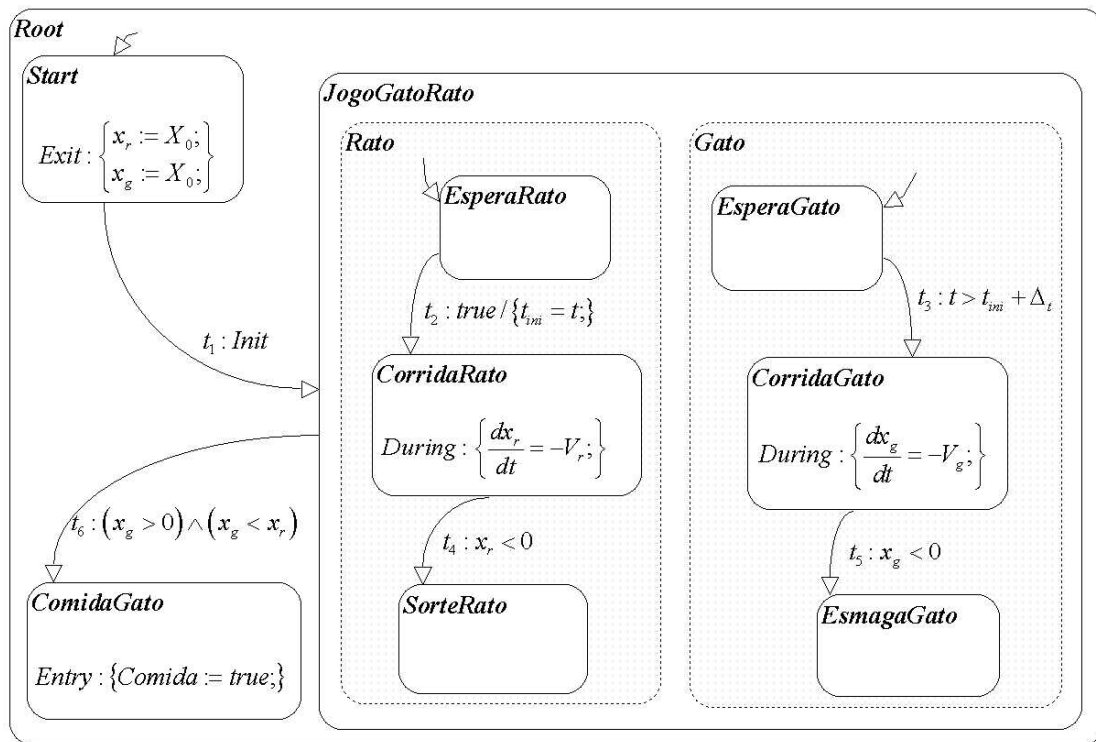


Figura 3.5 Statechart para descrição do jogo do gato e do rato

Este statechart representa uma competição entre um gato e um rato e foi apresentado inicialmente por Kesten em [Kesten92]. A sua simplicidade permite evidenciar algumas das potencialidades do formalismo dos Statecharts Híbridos para a modelação de sistemas com componentes contínuas e discretas. O exemplo pode ser descrito da seguinte forma: após o botão de *Start* ser premido, o rato começa a correr de uma determinada posição, em linha recta, directamente para um buraco que se

encontra na parede, e que dista  $X_0$  da posição inicial. O rato corre a uma velocidade constante  $V_r$ . Após um atraso temporal pré-definido,  $\Delta t$ , o gato é solto na mesma posição inicial e persegue o rato a uma velocidade constante,  $V_g$ , ao longo do mesmo percurso. Com a execução do statechart pode-se saber se o gato consegue apanhar o rato, ou se o rato consegue atingir o seu santuário e o gato se esmaga contra a parede. A especificação usa as variáveis contínuas  $x_r$  e  $x_g$  para medir a distância do rato e do gato, respectivamente, da parede.  $X_0$ ,  $V_r$ ,  $V_g$  e  $\Delta t$  são valores constantes. A evolução do statechart é iniciada com o botão de *Init* que provoca o abandono do estado *Start*. No instante da desactivação do estado *Start* é executada a acção associada à sua saída, na qual são iniciadas as variáveis de estado  $x_r$  e  $x_g$ . Inicia-se então o jogo com o rato a entrar imediatamente em competição, sinalizando entretanto o início da sua corrida, que é “sentido” pelo gato após um determinado atraso,  $\Delta t$ . Das velocidades do gato e do rato, e do atraso na partida do gato, dependerão a sorte do rato ou a refeição do gato.

### 3.1.6 A semântica de execução

Duas componentes da semântica dos Statecharts são normalmente consideradas: a parte estática e a parte dinâmica. A semântica estática é a descrição de constrangimentos estruturais que não podem ser capturados adequadamente através de descrições sintácticas. Por exemplo, a interpretação de transições e a detecção de transições inválidas são questões de semântica estática. Hong, em [Hong95], apresenta uma lista de regras para a implementação da semântica estática.

No presente trabalho, a implementação do formalismo dos Statecharts em Modelica obriga a algumas restrições para as transições:

- Uma transição não pode “atravessar” a linha delimitadora de nenhum estado, isto é, não são implementadas transições entre diferentes níveis hierárquicos.
- O disparo de uma transição não pode levar o statechart para uma configuração ilegal. Uma configuração ilegal pode ser atingida devido ao não determinismo. Isto pode ocorrer quando duas ou mais transições com o mesmo estado de origem são disparadas no mesmo instante.

A semântica dinâmica trata da evolução do statechart quando em execução. Cada passo deve ser definido com precisão incluindo todas as suas ramificações e efeitos colaterais. Um passo significa a realização de um conjunto de transições, sendo definido como o conjunto máximo de transições que são consistentes, compatíveis e relevantes na configuração inicial e disparadas pelos sinais de comunicação (sinais gerados pelas acções). Duas transições são consistentes quando pertencem a componentes paralelos. São compatíveis se, na acção de uma das transições, não existe nenhum sinal que apareça negado no evento da outra. Uma transição é relevante se o seu estado origem faz

parte da configuração. Quando disparada, uma transição envia sinais através das acções associadas, que podem, instantaneamente, habilitar novas transições (relevantes e compatíveis).

O comportamento de um sistema descrito por um statechart é um conjunto de execuções possíveis, cada uma representando a resposta do sistema a uma sequência de estímulos externos gerados pelo seu ambiente. Uma execução consiste numa série de momentos instantâneos de avaliação; cada momento é denominado *status*. O primeiro da sequência é o *status* inicial, e cada subsequente é obtido do seu predecessor através da execução de um passo.

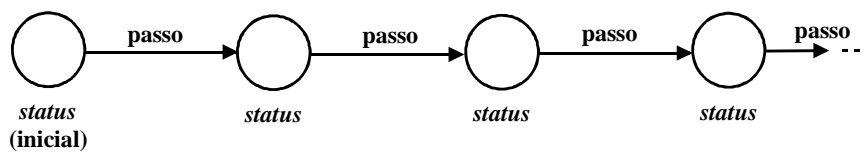


Figura 3.6 Evolução da execução de um statechart

O *status* contém informação sobre estados activos e actividades, valores de variáveis e condições, e eventos gerados. No início de cada passo, o ambiente fornece ao sistema em descrição os estímulos externos; estes estímulos, em conjunto com as mudanças que ocorrem no sistema, durante e após o passo precedente, são responsáveis pelo disparo de transições e tornam operacionais as actividades associadas aos estados. Como resultado disso, o sistema move-se para um novo *status*: alguns estados são desactivados e outros são activados, valores de condições e de variáveis são alterados, novos eventos são gerados, e a evolução prossegue de maneira semelhante.

Além de ser possível associar acções ao disparo de transições, elas também podem ser associadas à entrada e à saída de estados. As acções associadas à entrada de um estado são executadas no passo em que ocorre a entrada no estado, como se se tratasse de uma acção associada à transição de entrada no estado. Da mesma forma, as acções associadas à saída de um estado são executadas no passo em que o estado é desactivado, como se estas estivessem associadas à transição de saída do estado.

Define-se uma configuração do statechart como o conjunto máximo de estados que o sistema pode ter activos simultaneamente. A configuração básica refere-se ao conjunto máximo de estados básicos que um sistema pode ter activos simultaneamente. No exemplo da Figura 3.5, {*CorridaRato*, *EsperaGato*} é uma configuração básica e a sua configuração completa é o conjunto {*Root*, *JogoGatoRato*, *Gato*, *Rato*, *CorridaRato*, *EsperaGato*}. Num determinado momento, o conjunto de actividades operacionais nos estados de uma configuração definem o sistema DAE que tem de ser avaliado.



A configuração referida acima define o sistema de equações diferenciais:

$$\begin{cases} \dot{x}_r = -V_r \\ \dot{x}_g = 0 \end{cases}$$

Num passo, o sistema realiza tipicamente operações de quatro tipos: transições, actividades, acções realizadas na entrada de estados e acções realizadas na saída de estados. A execução de um passo deve sempre conduzir o sistema para uma configuração legal.

### 3.1.7 Especificação sintáctica formal

A sintaxe formal dos Statecharts Híbridos é seguidamente apresentada de forma simplificada [Turine98]. Um statechart híbrido, conforme é definido neste trabalho, é uma *tuple* com 12 elementos  $\langle S, r_s, f_s, l_s, d_s, V, C, E, A_c, A_t, R_t, T \rangle$ , na qual:

- $S = \{s_1, s_2, \dots, s_n\}; n \geq 1$  é um conjunto de estados.
- $r_s : S \rightarrow 2^S$  é a função de hierarquia, responsável pela definição dos sub-estados de cada estado na hierarquia do statechart.

Se  $a \in S, b \in S \wedge r_s(a) = r_s(b)$  então  $a = b$ ; não existem estados distintos com o mesmo conjunto de sub-estados.

$a \in S$  é básico quando  $r_s(a) = \emptyset$ , isto é, quando não contém sub-estados.

Se  $a \in S$ , tal que para  $\forall s \in S, a \notin r_s(s)$ , então  $a$  é o estado raiz do statechart, isto é, não possui nenhum estado ancestral na hierarquia.

Definem-se também, como extensões da função hierarquia, as funções  $r_s^*$  e  $r_s^+$  como sendo:

$$r_s^*(s) = \bigcup_{i \geq 0} r_s^i(s) \quad r_s^+(s) = \bigcup_{i \geq 1} r_s^i(s)$$

$r_s^i(s)$  representa o conjunto de estados descendentes de  $s$  no nível hierárquico  $i$ , logo

$$r_s^0(s) = \{s\}$$

$r_s^*(s)$  é, então, o conjunto de estados formado por  $s$  e todos os estados descendentes de  $s$  na hierarquia.  $r_s^*(r)$  é o conjunto de todos os estados da hierarquia, sendo  $r$  o estado raiz do statechart.

Se  $s_1$  e  $s_2$  são estados e  $s_1 \in r_s^*(s_2)$  então  $s_1$  é chamado descendente de  $s_2$  e  $s_2$  é chamado ancestral de  $s_1$ .

Para um conjunto de estados  $X \hat{=} S$  existe um único menor ancestral comum de  $X$ , designado por  $lca(X)$  (do inglês *least common ancestor*), e é definido da seguinte forma:  $lca(X) = x$  se  $X \hat{=} r_s^* \langle x \rangle$  e, para todo o  $s \hat{=} S$  tal que  $X \hat{=} r_s^* \langle s \rangle$ , leva a que  $x \hat{=} r_s^* \langle s \rangle$ .

–  $f_s : S \otimes \{AND, OR\}$  é a função decomposição hierárquica, definida para os estados não básicos.

Dois estados  $x$  e  $y$  são ortogonais, representados por  $x \wedge y$ , se  $x$  e  $y$  não estão relacionados ancestralmente e o seu  $lca$  é um estado AND.

Um conjunto de estados  $X$  é um conjunto ortogonal se, para todo o  $x, y \hat{=} X$ ,  $x = y$  ou  $x \wedge y$ .

Um conjunto de estados  $X \hat{=} S$  é chamado consistente se, para quaisquer dois estados  $x, y \hat{=} X$ , ou  $x$  e  $y$  estão relacionados ancestralmente ou  $x \wedge y$ . Um conjunto consistente diz-se ter consistência máxima se, para qualquer estado  $s \hat{=} S - X$ ,  $X \hat{=} \{s\}$  não é consistente. Chama-se configuração de um statechart a um sub-conjunto de  $S$  com consistência máxima. As configurações representam o estado global de um statechart. No exemplo da Figura 3.2 o conjunto  $\{B12, B22\}$  é um conjunto consistente, enquanto que  $\{Root, B, B1, B12, B2, B22, B3, B32\}$  é uma configuração, isto é, tem consistência máxima.

–  $l_s : H \otimes S$  é a função história responsável pela atribuição dos símbolos de história a estados do statechart. Somente os estados OR podem possuir história:  $l(H) = a$  se  $a \hat{=} S$  e  $f_s \langle a \rangle = \{OR\}$ .

–  $d_s : S \otimes S$  é a função *default* que é definida apenas para estados OR de tal forma que  $d_s \langle a \rangle \hat{=} r_s \langle a \rangle$ . Se  $b \hat{=} d_s \langle s \rangle$  então  $b \hat{=} d_s^+ \langle s \rangle$ .

Todos os estados *default* são activados quando os seus estados ancestrais directos forem activados (excepção para o caso do estado ancestral directo possuir história).

–  $V$  é o conjunto de expressões do statechart.

–  $C$  é o conjunto de condições do statechart. Uma condição pode ser uma combinação lógica de outras condições através dos operadores lógicos *and*, *or* e *not*, ou então pode ser uma expressão relacional entre variáveis do statechart.

–  $E = \{e_1, e_2, \dots, e_n\}$  é o conjunto finito de eventos do statechart;

$h \hat{=} E, h$  é o evento nulo.

Sendo  $E_p$  o conjunto de eventos primitivos do statechart, se  $e_1 \hat{=} E_p$ , então  $e_1 \hat{=} E$ .

A mudança do valor lógico de uma condição é um evento, bem como a mudança do valor de uma expressão.

Ocorre um evento sempre que um estado é activado ou desactivado.

Um evento pode ser composto pela combinação lógica de outros eventos.

Se  $e \in E$  e  $c \in C$  então  $e \wedge c \in E$ , isto é, um evento pode ser a combinação de um evento com uma condição.

–  $A_c = \{a_1, a_2, \dots, a_n\}$  é o conjunto de acções do statechart.

$s \in A_c$ ,  $s$ , é a acção nula.

A atribuição do valor de uma condição a uma condição primitiva é uma acção, bem como a atribuição do valor de uma expressão a uma variável é uma acção.

Um evento primitivo é uma acção, e uma acção composta por várias acções em sequência é também uma acção.

–  $A_t = \{A_{t1}, A_{t2}, \dots, A_{tm}\}$  é o conjunto de actividades do statechart.

Uma actividade define o comportamento do sistema no que diz respeito à sua evolução contínua dentro de um local particular, neste caso dentro de um estado do statechart. As actividades definem o (ou parte do) comportamento contínuo do statechart quando o respectivo estado está activo. Uma configuração define um sistema de equações diferenciais e algébricas com solução. Todas as diferentes configurações do statechart devem definir DAEs com solução.

–  $R_t = \{r_1, r_2, \dots, r_n\}$ , com  $R_t \subseteq E \times A$ , é o conjunto de rótulos das transições. A forma geral do rótulo é  $e \wedge c \vee a$ , onde  $e \in E$ ,  $c \in C$  e  $a \in A$ .

–  $T = \{t_1, t_2, \dots, t_n\}$ , com  $T \subseteq 2^S \times R \times 2^S$ , é o conjunto finito de transições.

Uma transição  $t = (x, r, y)$  é composta por um estado origem,  $x \in S$ , por um estado destino,  $y \in S$ , e por um rótulo,  $r \in R_t$ . É necessário que, para toda a transição  $t \in T$ , se  $in(t)$  for o estado origem e  $out(t)$  for o estado destino, então  $in(t)$  e  $out(t)$  têm de ter o mesmo ancestral directo,  $s$ , com  $f_s \langle s \rangle = OR$ , isto é,  $in(t) \in r_s^{-1} \langle s \rangle$  e  $out(t) \in r_s^{-1} \langle s \rangle$  (não são possíveis transições inter-nível).

Para o statechart da Figura 3.2 o conjunto de estados é  $S = \{Root, A, A_1, A_2, A_3, B, B_1, B_{11}, B_{12}, B_2, B_{21}, B_{22}, B_{23}, B_3, B_{31}, B_{311}, B_{312}, B_{32}\}$ , “Root” é o estado raiz e o conjunto de estados básicos é  $\{A_1, A_2, A_3, B_{11}, B_{12}, B_{21}, B_{22}, B_{23}, B_{311}, B_{312}, B_{32}\}$ . Por exemplo, a função de hierarquia para os estados “A” e “B” é, respectivamente,  $\rho_s(A) = \{A, A_1, A_2, A_3\}$ ,  $\rho_s(B) = \{B, B_1, B_2, B_3\}$ . A função  $\phi_s$  para os

mesmos estados é  $\phi_s(A) = \{OR\}$  e  $\phi_s(B_3) = \{AND\}$ . A função história é  $\lambda_s(H)=B_2$ . A função *default* para o estado “A” e “B” é, respectivamente,  $\delta_s(A) = \{A_1\}$  e  $\delta_s(B) = \{B_{11}, B_{21}, B_{31}\}$ . O conjunto de expressões de evento é  $E = \{e_1, e_2, e_3[c_3], e_4, e_5, e_6, e_7, e_8, e_9[c_9], e_{10}, e_{11}, e_{12}\}$  e o conjunto de transições é dado por  $T = \{(B, e_1, A), (A, e_2/e_9, B), (A_3, e_3[c_3], A_1), (A_1, e_4, A_2), (A_2, e_5, A_3), (B_{12}, e_6, B_{11}), (B_{11}, e_7, B_{12}), (B_{21}, e_8, B_{22}), (B_{22}, e_9[c_9]/e_{11}, B_{23}), (B_{23}, e_{10}, B_{22}), (B_{32}, e_{11}, B_{31}), (B_{31}, e_{12}, B_{32})\}$ .

### 3.1.8 Herança de comportamento nos Statecharts

No que diz respeito à herança de comportamento questiona-se [Harel97a]: como se deve relacionar um statechart A com um statechart B para assegurar alguma conformidade e encorajar a reutilização?

A linha principal seguida por Harel é basear os dois statecharts na mesma topologia estados/transições, de tal forma que B herda todos os estados e transições de A. Muito embora estes não possam ser removidos em B, algumas alterações são permitidas; os estados podem ser modificados de três formas:

- decompor um estado através de decomposições OR (em sub-estados) ou AND (em estados ortogonais);
- adicionar sub-estados a um estado OR;
- adicionar componentes ortogonais a qualquer estado.

A última forma é a mais importante já que é usada para enriquecer, em B, as capacidades de A em termos de comportamento. Também podem ser adicionadas transições ao statechart, bem como modificar parcialmente as transições herdadas. Por exemplo, se a transição for rotulada por **event[guard]action**, podem ser feitas mudanças ao nível do evento, na protecção da transição ou mesmo na lista de acções. Embora não seja permitido remover explicitamente uma transição, ela pode ser removida implicitamente fazendo a protecção falsa.

## 3.2 Implementação do formalismo dos Statecharts em Modelica

Alguns outros formalismos foram modelados com sucesso em linguagem Modelica. Citam-se, por exemplo, a modelação das Redes de Petri e das Máquinas de Estados Finitas [Mosterman98] ou a modelação do formalismo dos *Bond Graphs* [Broenink97]. Nesta secção relatam-se os procedimentos para a implementação do formalismo dos Statecharts, com características híbridas,

em linguagem Modelica. Nas secções seguintes são apresentados os métodos e as bibliotecas de modelos para a modelação do formalismo dos Statecharts, bem como um exemplo de utilização dessas bibliotecas para o caso simples do statechart da Figura 3.5.

Em Modelica são usadas DAEs para a descrição do comportamento dinâmico de sistemas físicos; estas equações poderão ser contínuas, descontínuas ou condicionais. A linguagem Modelica também oferece mecanismos para modelação de sistemas de eventos discretos ou sistemas amostrados. Para a geração do código que implementa o formalismo considera-se que um statechart é um modelo em linguagem Modelica e consiste numa composição de estados e transições. Esta abordagem foi seguida por Mosterman, em [Mosterman98], para a modelação do formalismo das Redes de Petri na linguagem Modelica.

Foram então desenvolvidos modelos em Modelica para os elementos básicos de um statechart: estados (*BasicState*, *State*, *OrthogonalState*, *RootState*) e transições (*Transition*). O modelo final do statechart é obtido por composição de modelos em Modelica ligados através de terminais estado-transição-estado.

A ideia subjacente à semântica dinâmica dos Statecharts em Modelica consiste em considerar uma avaliação do modelo como a execução de um passo na semântica dos Statecharts. Como todas as equações são avaliadas concorrentemente em Modelica e como, quando um evento ocorre, a integração contínua é interrompida, fica garantido que quaisquer acções associadas aos eventos do statechart somente se fazem sentir no sistema contínuo de equações quando a integração contínua é retomada.

### 3.2.1 Utilização da hipótese síncrona

Duas aproximações para a execução dos passos são normalmente consideradas: a hipótese assíncrona e a hipótese síncrona. Segundo Pnueli [Pnueli91], a hipótese síncrona assume que o sistema é infinitamente mais rápido que o ambiente e, então, a resposta a um estímulo externo introduzido num passo é sempre gerada no mesmo passo. De facto, a hipótese síncrona é uma abstracção que limita a interferência que pode ocorrer no período de tempo que separa o estímulo da sua resposta. Para a garantia da hipótese síncrona o modelo deve ser avaliado na sequência apresentada na Figura 3.7.

A hipótese síncrona para implementação dos Statecharts é naturalmente garantida pelo princípio síncrono dos sistemas híbridos em Modelica [Otter99], [Elmqvist00]. Num statechart, as actividades globais são modeladas por um conjunto de DAEs que descreve todo o comportamento contínuo da configuração na avaliação em causa. Quando ocorre um evento, detectado pelo

conjunto de funções de monitorização de eventos, a integração da DAE contínua é interrompida e assim todas as acções associadas aos eventos (incluindo os eventos gerados por acções) são sempre executadas antes da integração contínua ser retomada. Isto inclui o disparo de todas as transições, e a realização de todas as acções associadas às transições e à entrada e saída dos estados.

	Entidades avaliadas	Implementação em Modelica (comentários)
Avaliação do modelo ( $t$ )	Actividades ( $t- \varepsilon$ )	equações condicionais (conjunto de equações contínuas)
	disparo de transições	avaliação dos eventos e condições de guarda das transições
	desactivação de estados	suportado por ligações de desactivação hierárquicas
	activação de estados	suportado por ligações de activação hierárquicas
	acções das transições	equações <i>when clause</i> (podem ser usadas para iniciar eventos temporais)
	acções de entrada nos estados	equações <i>when clause</i> (podem ser valores iniciais para variáveis de estado das equações diferenciais contínuas definidas nas actividades dos estados)
	acções de saída dos estados	equações <i>when clause</i> (podem ser usadas para armazenar dados para a implementação do conceito de história)
<i>status</i> final da avaliação <i>n</i> disponível		
avaliação do modelo ( $t+ \varepsilon$ )	Actividades ( $t$ )	
	...	

Figura 3.7 Avaliação em *run-time* do statechart e a sua implementação em Modelica

A Figura 3.7 mostra como os componentes de uma avaliação do modelo podem ser suportados em Modelica. Por exemplo, as acções de entrada podem ser suportadas através de equações *when clause* e podem ser usadas para a atribuição de valores a variáveis de estado do conjunto de equações diferenciais, definidas como actividades no respectivo estado (exemplo da inicialização das variáveis  $x_r$  e  $x_c$  na entrada do estado *Start* do exemplo apresentado na Figura 3.5).

### 3.2.2 Implementação da hierarquia dos statecharts

Nos statecharts com hierarquia, os eventos de activação e desactivação devem ser propagados aos sub-estados. Existem duas possibilidades em Modelica:

i) Armazenar os sub-estados em listas de modelos e implementar a hierarquia à custa de listas ligadas de estados. A propagação dos eventos de activação/desactivação seriam suportados com algoritmos e ciclos repetitivos com os problemas inerentes de indexação e inicialização de listas com estados de diferentes tipos. Porém, esta hipótese não é, à presente data, implementável pois o actual compilador Modelica não o permite.

ii) Suportar a hierarquia com ligações auxiliares de hierarquia (Figura 3.8). Todos os estados são inicializados da mesma forma, sendo somente necessário estabelecer as ligações hierárquicas adequadas. Quando ocorrem os eventos de activação/desactivação, estes são suportados por equações de variáveis Booleanas, não havendo necessidade de usar algoritmos. Isto permite a propagação instantânea dos eventos de activação/desactivação a todos os estados pertencentes à hierarquia. A direcção de propagação única dos eventos, níveis hierárquicos superiores (estados ancestrais) para níveis hierárquicos inferiores (estados descendentes) é suportada por dois terminais (*input*, *output*) em cada estado, excepto no estado raiz.

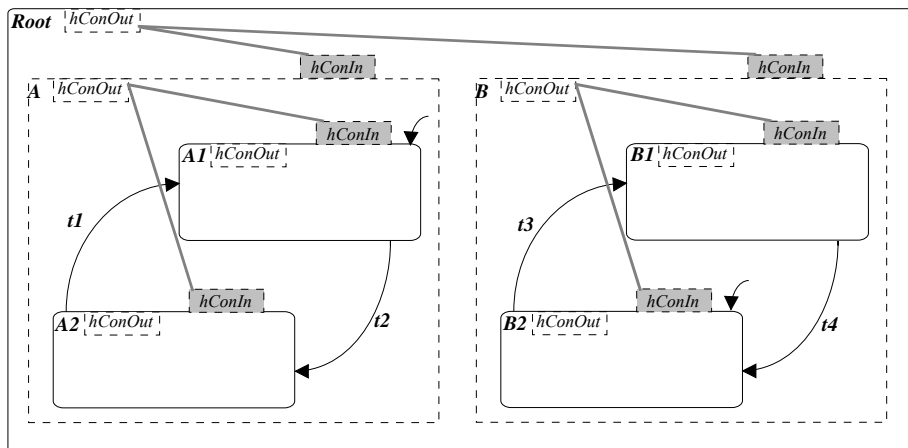


Figura 3.8 Ligadores hierárquicos para propagar eventos a sub-estados

### 3.2.3 Dois níveis de implementação

Na implementação dos Statecharts em Modelica foram considerados dois níveis de implementação: biblioteca dos Statecharts e modelos dos componentes a modelar.

A biblioteca dos Statecharts inclui os seguintes modelos:

- Modelo para representar as ligações hierárquicas estado a estado: **HierarchicalConnection**
- Modelos para ligação de transições a estados: **FireCut**, **SetCut**
- Modelos para a representação de estados: **BasicState**, **OrthogonalState**, **State**, **RootState**
- Modelos para a representação de transições: **Transition**

Os modelos da biblioteca dos Statecharts são responsáveis pela captura dos eventos relacionados com o disparo de transições e com a activação e desactivação de estados, devendo estes efectuar-se quando a transição é disparada. Os modelos da biblioteca têm também que codificar a activação e desactivação de sub-estados, de modo a implementar a semântica dos estados OR e AND.

A Figura 3.9 define os modelos onde são habilitadas a execução de acções e a operacionalidade das actividades do statechart durante uma avaliação do modelo.

	Tipo	Implementação em Modelica
<b>Acções</b> (Modelação do comportamento discreto e dos eventos)	<i>Transition action</i>	Modelo <i>Transition</i> Habilitação da acção da transição e desactivação do estado origem se se der o disparo
	<i>Entry action</i>	Modelo <i>State</i> Activação de todos os estados ortogonais (estado AND); activação do estado <i>default</i> ou do mecanismo de história (estado OR)
	<i>Exit action</i>	Modelo <i>State</i> Desactivação de todos os sub-estados
<b>Actividades</b> (Modelação do comportamento contínuo)	<i>During activity</i>	Modelo <i>State</i> Habilitação das actividades correspondentes

Figura 3.9 Implementação em Modelica do suporte à execução de acções e habilitação de actividades

O código Modelica para o modelo do componente, cujo comportamento é modelado pelo statechart, pode ser gerado automaticamente a partir da especificação do statechart e inserido na classe representativa do modelo do componente. Sendo assim, o utilizador final só terá que definir graficamente o statechart, em editor gráfico adequado, definir todos os eventos, ligar graficamente as transições aos estados, definir as acções e actividades e, depois, associar o statechart a uma classe em Modelica.

A Figura 3.10 mostra as construções em Modelica que devem ser usadas para implementação das acções e actividades.

	Tipo	Implementação em Modelica (modelo do componente)
<b>Acções</b> (Modelação do comportamento discreto e dos eventos)	<i>Transition action</i>	<pre>when transitionName.action then //equações ou algoritmos da acção end when</pre>
	<i>Entry action</i>	<pre>when stateName.activate then //equações ou algoritmos da acção end when</pre>
	<i>Exit action</i>	<pre>when stateName.deactivate then //equações ou algoritmos da acção end when</pre>
<b>Actividades</b> (Modelação do comportamento contínuo)	<i>During activity</i>	<pre>if stateName.active then // conjunto de equações diferenciais e algébricas end if</pre>

Figura 3.10 Descrição e implementação das acções e actividades ao nível do componente



### 3.2.4 Mecanismos para a implementação da história

Algumas implementações dos statecharts usam os chamados mecanismos de história. A história num estado de um statechart, identificada pelo símbolo H, representa dinamicamente o sub-estado activo, da última vez que o estado ancestral directo (com o símbolo de história) foi desactivado.

Na presente implementação, a variável *lastState* do modelo *State* (ver o modelo *State* na secção seguinte) memoriza o *status* do estado imediatamente antes da sua desactivação (devido a um evento de desactivação hierárquica). Esta variável *lastState* deve ser inicializada com o parâmetro *defaultState* e é avaliada quando ocorre um evento de activação hierárquica. Para se usar o mecanismo de história, o parâmetro *history* do tipo Boolean, definido no modelo *State*, deve ser inicializado verdadeiro em todos os sub-estados do estado OR em causa. Assim, este activar-se-á quando receber um evento de activação hierárquica proveniente do seu estado ancestral.

### 3.2.5 Bibliotecas de modelos

Esta secção apresenta um conjunto de modelos, desenvolvidos no âmbito do presente trabalho, para a implementação do formalismo dos Statecharts em linguagem Modelica. Até ao momento da realização deste trabalho, a implementação disponível da linguagem Modelica, especificação 1.4, ainda não contempla listas de componentes; por isso, foram desenvolvidos modelos de estados com várias transições de saída e de entrada.

Na implementação proposta são usados dois tipos de terminais para criar ligações entre estados: terminais para estabelecer ligações estado-transição-estado e terminais usados na definição da hierarquia de estados do statechart.

Os modelos de terminais, para estabelecer ligações estado-transição-estado, são idênticos aos definidos por Mosterman, em [Mosterman97], para a implementação do formalismo das Redes de Petri em Modelica.

As classes *FireCut* e *SetCut* representam, respectivamente, terminais para ligação de transições ao seu estado origem e ligações ao seu estado destino, Figura 3.11. A ligação através de terminais do tipo *FireCut* permite transportar o *status* do estado origem para a transição (sinal *state*) e a sinalização do *status* da transição (sinal *fire*) para o estado destino.

```
connector FireCut
  Boolean state;
  Boolean fire;
end FireCut;
```

A ligação *SetSCut* é usada para sinalizar o estado destino do disparo de uma transição de entrada (sinal *set*). Este sinal permite a activação do estado destino.

```
connector SetCut
  Boolean set;
end SetCut;
```

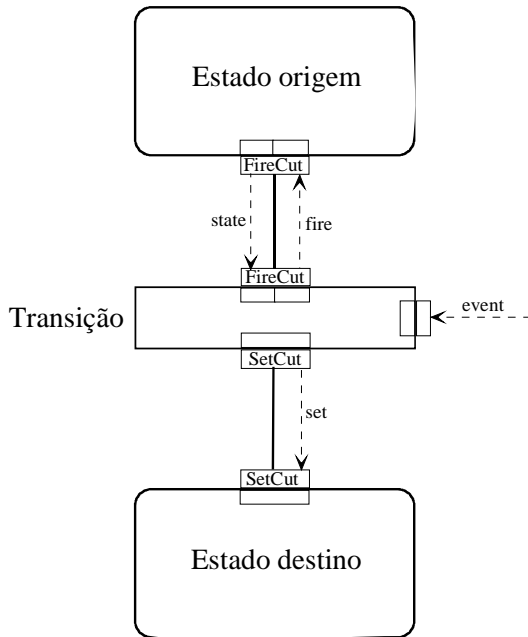


Figura 3.11 Uniões para ligação duma transição ao estado origem e ao estado destino

Os terminais que estabelecem ligações hierárquicas entre estados (definição da hierarquia) são representados pela classe *HierarchicalConnection*. Estas ligações permitem o envio de eventos de activação e desactivação hierárquica de estados.

```
connector HierarchicalConnection
  Boolean act;
  Boolean deact;
end HierarchicalConnection;
```

A classe *BasicState* é a classe base para representação de estados. Define as características estruturais e funcionais comuns a todos os tipos de estados. Utiliza variáveis do tipo *Boolean* para sinalizar a sua activação e desactivação e, também, a operacionalidade da sua actividade. Possui igualmente terminais para activação e desactivação hierárquica.

```

model BasicState
  Boolean activate, deactivate, active;
  HierarchicalConnection hConIn;
  HierarchicalConnection hConOut;
equation
  hConOut.act = activate;
  hConOut.deact = deactivate;
  active = (pre(active) or activate) and not deactivate;
end BasicState;

```

O modelo *Transition* representa uma transição. Contem terminais para ligação ao estado origem e ao estado destino, e usa variáveis do tipo **Boolean** para guardar o status do seu evento e da sua protecção, e também para sinalizar que a sua acção pode ser executada.

```

model Transition
  FireSCut inp;
  SetSCut out;
  Boolean event, guard (start = true), action;
equation
  out.set = inp.state and event and guard;
  inp.fire = out.set;
  action = out.set;
end Transition;

```

Se for activada a história, o estado deve recordar o seu último *status* (activo ou inactivo). Depois, na próxima entrada da activação do seu estado ancestral directo (estado pai), de que é notificado pelo seu terminal de activação hierárquica, o presente estado será activado. A variável do tipo **Boolean**, *lastState*, guarda o *status* anterior do estado quando é usada história.

Um estado pode ser activado quando se dá o disparo de uma transição de entrada ou quando o seu estado pai for activado. O sinal *activate* sinaliza que a acção associada à entrada num estado pode ser executada. Um estado pode ser desactivado quando uma transição de saída é activada ou quando o seu estado pai for desactivado. O sinal *deactivate* sinaliza que a acção associada à saída do estado pode ser executada. Na desactivação hierárquica, e quando existe história, a variável *lastState* é actualizada. Somente quando eventos de desactivação hierárquica ocorrem é necessário guardar o último *status* do estado. É também necessário sinalizar o *status* do estado a transições de saída do estado.

```

model State
  extends BasicState;
  parameter Boolean defaultState = false, history = false;
  SetCut inp;
  FireCut out;
  Boolean lastState (start = defaultState);
equation
  activate = if history then pre(inp.set) or (hConIn.act and lastState)
             else pre(inp.set) or (hConIn.act and defaultState);
  deactivate = pre(out.fire) or hConIn.deact;
  when (hConIn.deact and history) then
    lastState = pre(active);
  end when;
  out.state = active;
end State;

```

O modelo *State* representa um estado com uma transição de entrada e uma transição de saída. Foram também desenvolvidos outros modelos para representar estados com várias transições de entrada e/ou saída. Por exemplo, os modelos seguintes representam modelos com duas transições de entrada e uma transição de saída e vice-versa.

```

model StateSIn2Out1
  extends BasicState;
  parameter Boolean defaultState = false;
  parameter Boolean history = false;
  SetSCut inp1, inp2;
  FireSCut out;
  Boolean lastState (start = defaultState);
equation
  activate = if history then pre(inp1.set) or pre(inp2.set) or (hConIn.act and lastState)
             else pre(inp1.set) or pre(inp2.set) or (hConIn.act and defaultState);
  deactivate = pre(out.fire) or hConIn.deact;
  when (hConInp.deact and history) then
    lastState = pre(active);
  end when;
  out.state = active;
end StateSIn2Out1;

```

```

model StateSIn1Out2
  extends BasicState;
  parameter Boolean defaultState = false;
  parameter Boolean history = false;
  SetSCut inp;
  FireSCut out1, out2;
  Boolean lastState (start = defaultState);
equation
  activate = if history then pre(inp.set) or (hConIn.act and lastState)
            else pre(inp.set) or (hConIn.act and defaultState);
  deactivate = pre(out1.fire) or pre(out2.fire) or hConIn.deact;
  when (hConInp.deact and history) then
    lastState = pre(active);
  end when;
  out1.state = active;
  out2.state = active;
end StateSIn1Out2;

```

Um estado ortogonal, representado pela classe *OrthogonalState*, é o estado descendente directo (estado filho) de um estado AND. Este tipo de estado não permite propriedades de história ou estado *default* pois será sempre activado quando o seu estado pai for activado. Além disso, na presente sintaxe, também não são permitidas transições.

```

model OrthogonalState
  extends BasicState;
equation
  activate = hConIn.act;
  deactivate = hConIn.deact;
end OrthogonalState;

```

O estado raiz, representado pela classe *RootState*, só gera eventos de activação e desactivação hierárquica no início e no final da execução do statechart, uma vez que está sempre activo.

```

model RootState
  HierarchicalConnection hConOut;
equation
  hConOut.act = initial();
  hConOut.deact = terminal();
end RootState;

```

### 3.2.6 Utilização da biblioteca de modelos

A utilização da biblioteca de modelos apresentada na secção anterior é aqui exemplificada com a implementação do statechart da Figura 3.5. As acções necessárias para um uso correcto da biblioteca enumeram-se a seguir:

- declarar todos os parâmetros, constantes e variáveis;
- incluir todas as transições do statechart;
- incluir e inicializar todos os estados do statechart;
  - inicializar, para cada estado, as ligações hierárquicas;
  - inicializar as transições de entrada para cada estado;
  - inicializar as transições de saída para cada estado;
- estabelecer todas as ligações entre estados e transições;
- definir todos os eventos do statechart;
- definir todas as condições de protecção das transições que o statechart deva respeitar;
- definir todas as acções e actividades do statechart;
  - definir todas as acções associadas com as transições;
  - definir todas as acções de entrada e saída de estados;
  - definir todas as actividades associadas com os estados.

O exemplo apresentado abaixo em código Modelica (note-se que está incompleto) ilustra o uso da biblioteca de modelos para a implementação do statechart apresentado na Figura 3.5.

O modelo *Jogo* foi simulado usando a plataforma Dymola ([www.dynasim.se](http://www.dynasim.se) 2003) com diferentes velocidades e atrasos. Os resultados, na Figura 3.12, mostram a evolução dinâmica do statechart durante 5 segundos, para a configuração de velocidades ( $V_r = 2 \text{ m/s}$ ;  $V_g = 5 \text{ m/s}$ ) e  $\Delta t = 2 \text{ s}$ . Pode constatar-se que o gato começa a sua refeição (sinalizada pela variável *Meal*) 1.4 segundos após começar a sua corrida. Na Figura 3.12 estão também ilustradas as actividades dos estados *CorridaGato* e *CorridaRato*.

```

model Jogo
  parameter Real Delay = 2, Real X0 = 10, Real Vr = 2, Real Vg = 5;
  RootState Root;
  State Start (defaultState = true), State JogoGatoRato, ...;
  OrthogonalState Gato, Rato;
  Transition T1, T2, ...;
  Real xr, xg, tini;
  Boolean Init, Comida;
equation
  connect(Start.out, T1.inp);
  connect(JogoGatoRato, T1.out);
  connect(Root.hConOut,Start.hConInp);
  connect(Root.hConOut,JogoGatoRato.hConInp);
  connect(Root.hConOut,ComidaGato.hConInp);
  Init = time > 1;
  T1.event = Init;
  T3.event = time > tini + Delay;
  ...
  when T2.action then
    tini = time;
  end when;
  when Start.deactivate then
    reinit(xg , X0);
    reinit(xr , X0);
  end when;
  when ComidaGato.activate then
    Comida = true;
  end when;
  if CorridaRato.active then
    der (xr) = -Vr;
  else
    der(xr) = 0;
  end if;
  if CorridaGato.active then
    der (xc) = -Vc;
  else
    der(xc) = 0;
  end if;
end Jogo;

```

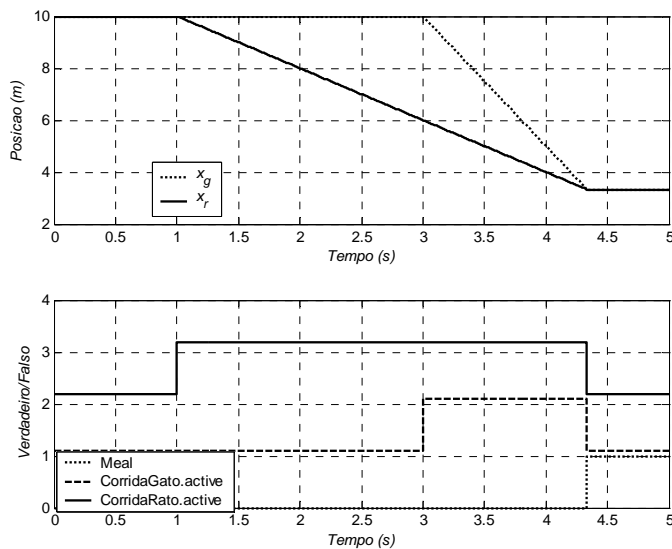


Figura 3.12 Execução do statechart da Figura 3.5 (codificado em Modelica) na plataforma Dymola

### 3.3 Metodologia de modelação proposta

A metodologia proposta para o desenvolvimento de bibliotecas de modelos de componentes, para uso em simulação dinâmica de sistemas hidráulicos, possui duas vertentes: metodologia para a modelação e metodologia para a organização de bibliotecas de modelos.

#### 3.3.1 Metodologia para a modelação

A metodologia proposta para a modelação considera que cada modelo é composto por duas perspectivas complementares: a sua estrutura e o seu comportamento.

A **estrutura** engloba toda a parte estática do modelo, os seus parâmetros, os seus terminais de ligação, componentes, etc... A utilização de uma filosofia orientada por objectos é assumida como a mais adequada para modelar a estrutura de sistemas complexos. De facto, o conceito de classe e de herança permite a estruturação dos modelos de uma forma hierárquica, conceito este comumente aceite como extremamente poderoso na modelação de sistemas complexos. Outro mecanismo importante na estruturação de modelos de sistemas é a interligação de modelos simples no sentido de modelar sistemas mais complexos. Este mecanismo permite reproduzir a topologia dos sistemas nos seus modelos. Por exemplo, um circuito eléctrico, contendo diferentes componentes eléctricos, pode ver a sua topologia reproduzida num modelo contendo modelos dos diferentes componentes do circuito, interligados da mesma forma. Neste caso, modelos de componentes isolados, como



resistências, condensadores, bobinas, etc., seriam usados na modelação de circuitos com vários e diferentes componentes. Para a modelação e simulação de sistemas físicos complexos, através da interligação de modelos mais simples, faz sentido que cada modelo de um determinado elemento físico permita ligações a outros elementos. Para poder ser interligado o modelo do componente necessitará de terminais de ligação.

O **comportamento** modela a parte dinâmica do modelo, isto é, a evolução do seu desempenho ao longo do tempo. Esse comportamento é influenciado por diversos factores, nomeadamente o próprio tempo, eventos captados ou alterações nos seus atributos, provenientes dos seus terminais de ligação.

A metodologia de modelação proposta usa o formalismo dos Statecharts embebido na linguagem Modelica. A Figura 3.13 mostra o processo da modelação de um determinado componente físico, segundo a presente proposta de modelação; a geração do código, em linguagem Modelica, pode ser dividida em duas partes complementares: o modelo estrutural é descrito através de classes em Modelica, enquanto que o comportamento é descrito através de equações (com as descontinuidades apropriadas) e, em seguida, inserido na classe correspondente.

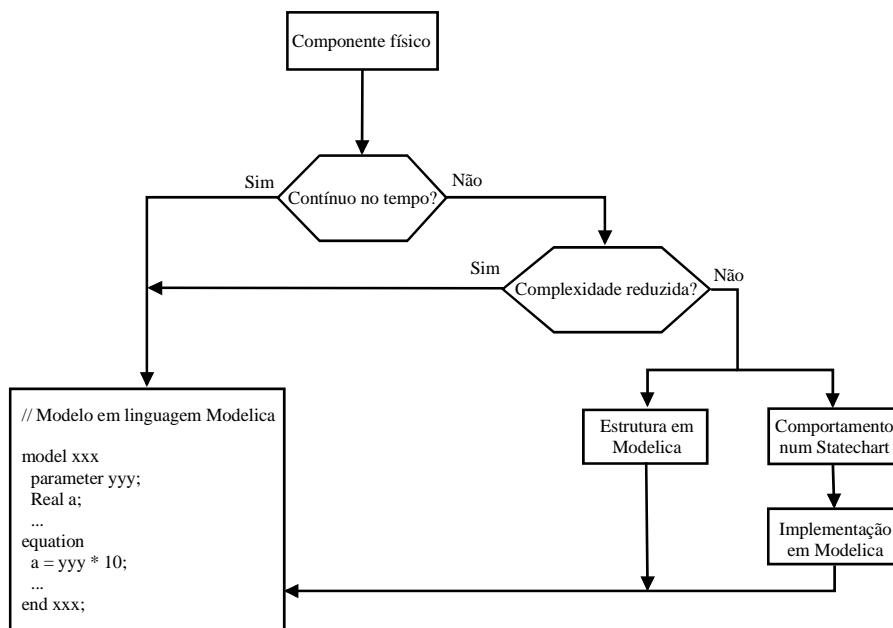


Figura 3.13 Metodologia para modelação de um componente físico isolado

Para modelos de sistemas que se comportam de uma forma contínua no tempo, a descrição do comportamento através de equações algébricas e diferenciais poderá ser feita directamente em linguagem Modelica sem haver a necessidade de associar um *statechart* ao modelo estrutural. Para outros modelos, há que avaliar a complexidade do seu comportamento e decidir se a associação de um *statechart* para descrição do comportamento do sistema, ou parte dele, é vantajosa. Resumindo,

comportamentos contínuos são modelados através de DAEs directamente em linguagem Modelica; comportamentos híbridos são modelados usando os mecanismos nativos da linguagem: directamente (comportamentos simples) ou através do formalismo dos Statecharts (comportamentos complexos).

O formalismo dos Statecharts pode também ser usado com uma base interessante de herança de comportamentos, tornando possível a redefinição, a extensão ou o refinar destes comportamentos e, assim, implementar modelos de diferentes complexidades para um mesmo componente. Este conceito, bem como a sua forma de implementação, é ilustrado através do exemplo seguinte, no qual um statechart é herdado e refinado.

Considere-se o exemplo de uma válvula de segurança (ver secção 2.3.3). Da hierarquia de herança estrutural e comportamental para a válvula de segurança, ilustrada na Figura 3.14, depreende-se a forma como o formalismo dos Statecharts é associado com a linguagem Modelica e os dois tipos de herança possíveis. A herança estrutural, natural de uma linguagem de modelação orientada por objectos, como é o caso da linguagem Modelica, e a herança de comportamentos híbridos através dos mecanismos de herança definidos por Harel [Harel97a] para o formalismo dos Statecharts.

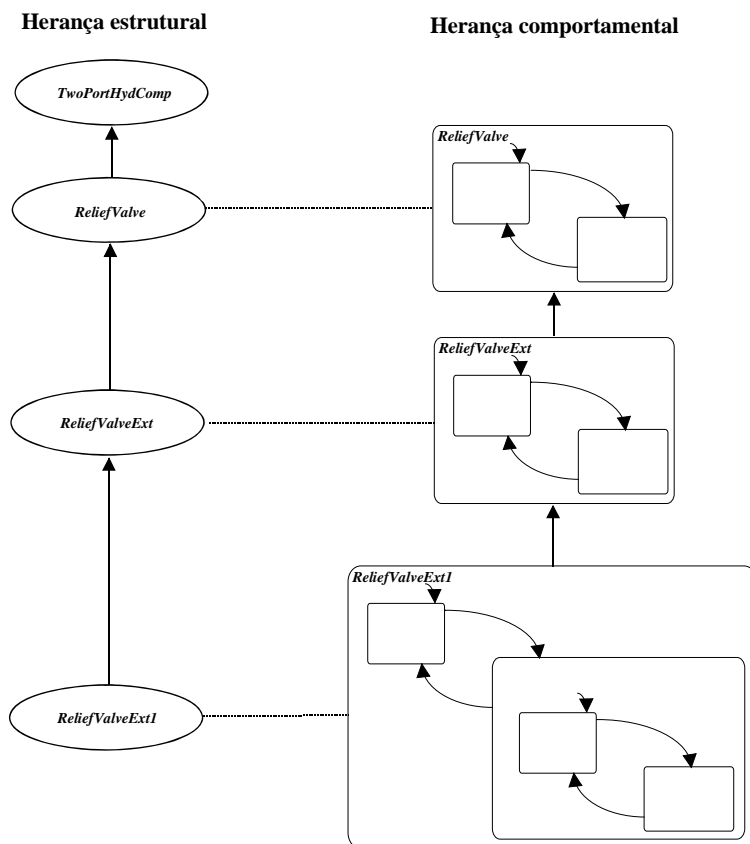


Figura 3.14 Hierarquia de herança para o modelo da válvula de segurança hidráulica

Os modelos da estrutura herdam a estrutura do modelo parcial, definido na secção 2.3.3, para componentes hidráulicos com dois terminais de ligação. O modelo comportamental mais simples da válvula considera que esta possui duas posições possíveis. Então, usando o formalismo dos Statecharts, o comportamento dinâmico pode ser representado por um statechart OR (*ReliefValve*), Figura 3.15, que contém dois sub-estados (*Aberta*, *Fechada*) e duas transições ( $t_A$ ,  $t_F$ ). O comportamento descrito pelo statechart pode ser interpretado da forma que a seguir se descreve. Inicialmente a válvula está fechada (o sub-estado *Fechada* é o estado *default*); em todos os passos de simulação a actividade,  $q_r = 0$ , do estado *Fechada* é avaliada; esta actividade modela o comportamento da válvula quando a válvula está fechada; se a válvula está totalmente fechada e a diferença de pressão aos terminais da válvula ( $\Delta p$ ) excede a pressão necessária para abrir a válvula (neste caso,  $pF$ ) então a transição  $t_A$  é disparada e o estado *Aberta* é activado; a actividade do estado *Aberta*,  $q_r = (Dp - pF) \times g_{op} + Dp \times g_{lk}$ , é então avaliada descrevendo o comportamento da válvula quando está aberta; esta actividade permanece operacional enquanto a transição  $t_F$  não for disparada; o evento desta transição detecta quando a pressão  $\Delta p$  é menor que a pressão definida pelo parâmetro  $pF$ , pressão mínima para manter a válvula aberta. Note-se que  $\Delta p$  é calculada em *TwoPortHydComp* e herdada pelo modelo *ReliefValve*.

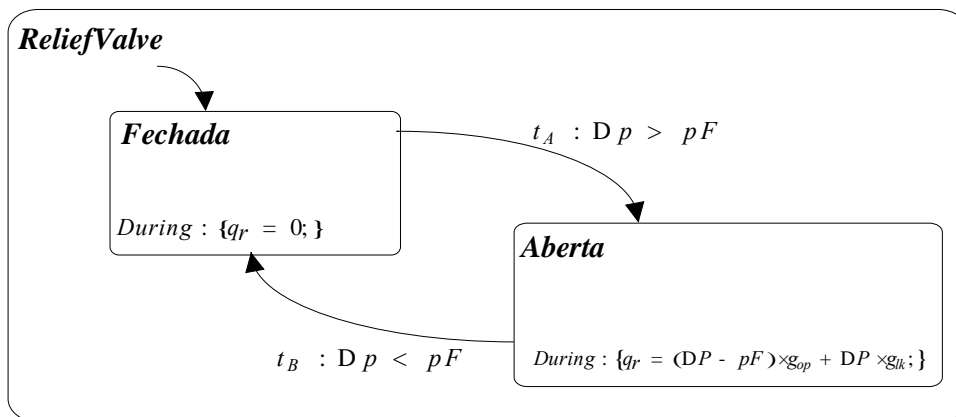


Figura 3.15 Statechart para descrição do comportamento do modelo **ReliefValve**

Os atributos  $q_r$  (caudal através do componente) e  $\Delta p$  (diferença de pressão aos terminais do componente) são herdadas do modelo *TwoPortHydComp*.

Num outro modelo, apresentado na Figura 3.16, pode melhorar-se o modelo anterior, considerando que a válvula apresenta fugas quando está fechada. No statechart *ReliefValveExt* redefiniu-se a actividade do estado *Fechada* de modo a que o caudal de fugas seja  $q_r = Dp \times g_{lk}$ .

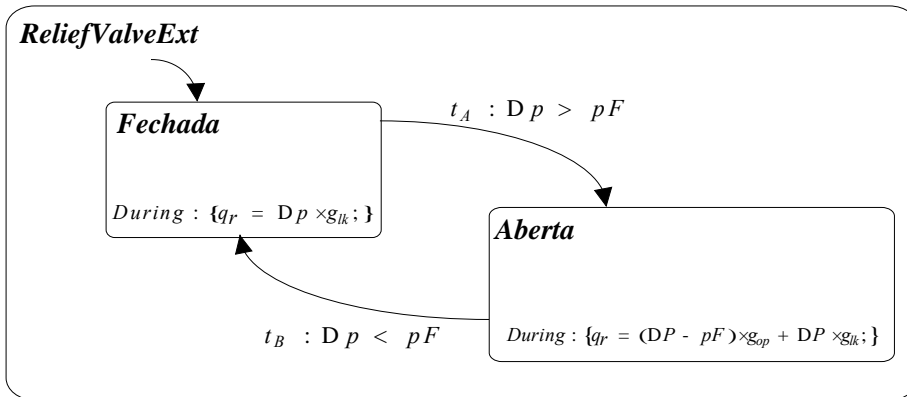


Figura 3.16 Statechart para descrição do comportamento do modelo *ReliefValveExt*

Se se pretender melhorar o comportamento do modelo representativo da válvula de segurança, considerando que deve ser modelado o comportamento durante a abertura da válvula, pode então usar-se o statechart da Figura 3.17, que corresponde a uma extensão do statechart da Figura 3.16. Aqui o estado *Aberta* foi redefinido, sendo agora um estado OR com os sub-estados *ParcialmenteAberta* e *TotalmenteAberta*. Considera-se que a válvula se comporta de uma forma linear na gama de operação, estado *ParcialmenteAberta*, e, quando está totalmente aberta, tem um comportamento do tipo orifício com secção constante.

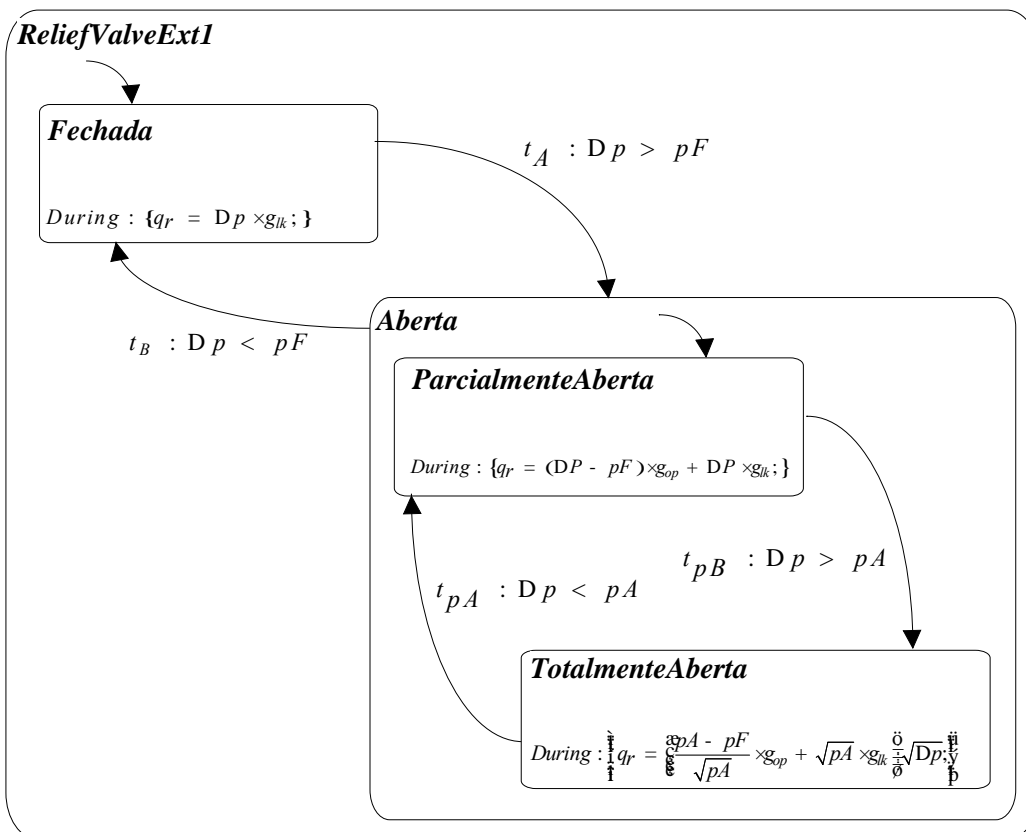


Figura 3.17 Statechart para descrição do comportamento do modelo *ReliefValveExt1*

A Figura 3.18 mostra os comportamentos para os três modelos da válvula quando sujeitos à mesma variação de pressão aos seus terminais.

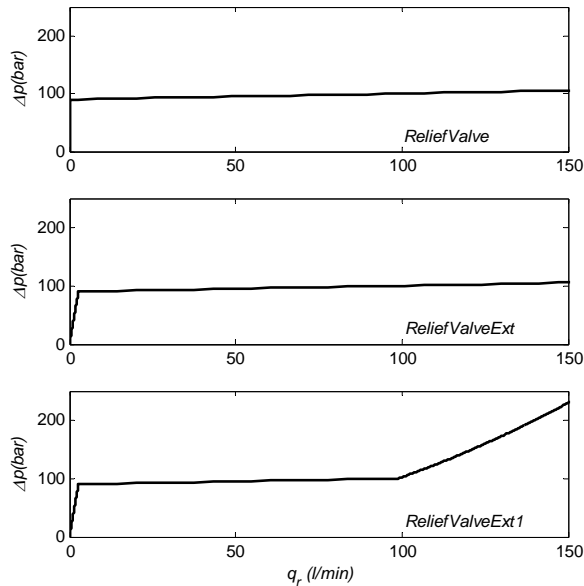


Figura 3.18 Curvas  $\Delta p$  versus  $q_r$  para o modelo ReliefValve e ReliefValveExt e ReliefValveExt1

### 3.3.2 Metodologia para organização de bibliotecas de modelos

O principal objectivo da organização de bibliotecas de modelos é a possibilidade de existirem múltiplos modelos para um mesmo componente. Esses modelos podem ser fruto de diferentes abordagens na modelação (modelos empíricos, semi-empíricos ou modelos baseados exclusivamente nas leis da física) e/ou também serem representados com diferentes níveis de complexidade. Desta forma conseguem-se diferentes desempenhos computacionais, permitindo a sua utilização em diferentes plataformas de hardware de simulação. Este conceito permite ajustar a complexidade do modelo global quando, por exemplo, se pretende simular o modelo em tempo real, no hardware disponível. Prevê-se então que modelos com complexidade menor possam ser simulados em tempo real em plataformas de mais baixo custo.

Esta forma de organização de modelos implica que todos os modelos de um determinado componente sejam definidos com a mesma interface, isto é, os mesmos terminais de ligação. De facto, só mantendo a estrutura de ligação é possível uma rápida e eficaz substituição de modelos. Numa linguagem orientada por objectos essa interface pode facilmente ser herdada de um modelo abstracto que capte as características comuns aos diferentes modelos. Por exemplo, pode-se considerar que uma electroválvula possui quatro terminais hidráulicos e um terminal eléctrico; se

essa válvula estiver inserida num circuito hidráulico, o facto da descrição do seu comportamento variar não afecta a estrutura do circuito hidráulico, muito embora possa variar a DAE (e a sua dimensão) ou o statechart que define o comportamento dinâmico da válvula.

Do ponto de vista de implementação dos conceitos acima descritos afiguram-se três possibilidades com suporte na linguagem Modelica.

A primeira possibilidade, representada em exemplo no código seguinte, define num mesmo modelo as diferentes complexidades, fornecendo um índice de selecção para indicar qual a DAE ou o statechart a usar na simulação. De notar que o modelo *A* herda a interface *interModel*. A utilização do modelo, no modelo *Use*, permite seleccionar a complexidade desejada.

Esta possibilidade deve ser utilizada quando não existam, ou não se antevêa que venha a existir, um número elevado de modelos de complexidades diferentes para um dado componente.

```
model A
  extends interModel;
  parameter Integer compIndex = 1;
  ...
equation
  if compIndex == 1 then
    DAE1 e/ou Statechart1
  else if compIndex == 2 then
    DAE2 e/ou Statechart2
  end if;
end A;

model Use
  A useA(compIndex=1); //ou compIndex = 2
  ...
equation
  ...
end Use;
```

A segunda possibilidade sugere a herança do parâmetro que caracteriza a complexidade do modelo, *compIndex*, com a definição de novos comportamentos em modelos de níveis inferiores na hierarquia de herança. Neste caso, seguindo o código do exemplo, o modelo *AI* herda o índice de complexidade, *compIndex*, do modelo *A*, definindo um novo comportamento ao qual é atribuído um índice de complexidade 2. A utilização, no modelo *Use*, permite seleccionar o comportamento

definido em  $A$  (complexidade de índice 1) ou em  $A1$  (complexidade de índice 2). Com esta estratégia é possível definir um grande número complexidades, sem complicar os modelos, já que cada nova complexidade é implementada num novo modelo.

```

model A
  extends interModel;
  parameter Integer compIndex = 1;
  ...
equation
  if compIndex == 1 then
    DAE1 e/ou Statechart1
  end if;
end A;

model A1
  extends A (compIndex = 2);
  ...
equation
  if compIndex == 2 then
    DAE2 e/ou Statechart2
  end if;
end A1;

model Use
  A1 useA(compIndex=1); //ou compIndex = 2
  ...
equation
  ...
end Use;

```

A terceira possibilidade emana do conceito de parametrização de classes para a substituição de modelos com a mesma interface. Nesta aproximação, e seguindo o exemplo apresentado em código, são definidos diferentes modelos com a mesma interface (herança da interface definida em *interModel*). No modelo *Use* é usado o modelo *A* que corresponde a uma determinada complexidade. A instanciação da classe *A* (elemento *useA*) com a palavra reservada *replaceable* permite a sua re-declaração em classes descendentes de *Use*. Assim *useA*, componente do modelo *Use1*, passará a ser do tipo *B* (onde está definida outra complexidade para o componente). Esta

possibilidade não define, de forma explícita, um índice de complexidade. É, no entanto, possível fazer uma selecção nominativa dum modelo, com uma complexidade adequada, para utilização numa determinada experiência.

```
model A
  extends interModel;
  ...
equation
  DAE1 ou Statechart1
end A;

model B
  extends interModel;
  ...
equation
  DAE2 ou Statechart2
end B;

model Use
  replaceable A useA;
  ...
equation
  ...
end Use;

model Use1
  extends Use (redeclare B useA);
  ...
equation
  ...
end Use1;
```

Em resumo, a metodologia de modelação proposta considera a utilização de uma linguagem de modelação orientada por objectos em conjunto com um formalismo gráfico para a modelação de sistemas físicos complexos. Também se propõe uma metodologia para a organização de bibliotecas de modelos, onde poderão coexistir modelos de diferentes complexidades para um mesmo componente, que facilite a selecção da complexidade adequada à realização de diferentes tipos de simulação (simulação em tempo não real e simulação em tempo real). Das três possibilidades, anteriormente descritas, a referida em terceiro lugar foi a mais extensamente usada neste trabalho para a organização dos modelos desenvolvidos.





## 4 MODELAÇÃO DE SISTEMAS HIDRÁULICOS

A aplicação da metodologia de modelação, definida no Capítulo 3, ao desenvolvimento de bibliotecas de modelos pressupõe a existência de vários modelos, que poderão ter diferentes complexidades, para um mesmo componente. Neste contexto, são propostos, neste capítulo, modelos vários para alguns componentes necessários à modelação de sistemas hidráulicos. São apresentados, por exemplo, modelos para o óleo hidráulico, modelos de actuadores lineares e modelos de atrito. Os modelos desenvolvidos para modelação de válvulas proporcionais são apresentados no capítulo seguinte.

### 4.1 Fluidos hidráulicos e suas características

O fluido hidráulico é o meio de transmissão de energia em sistemas hidráulicos. Os fluidos hidráulicos modernos existentes no mercado resultam de fórmulas complexas (que estão fora do âmbito deste trabalho) cujo objectivo é garantir certos requisitos de compressibilidade, viscosidade e massa específica (ou densidade), mantendo-os estáveis durante mais tempo. As características mais importantes dos fluidos, no que diz respeito à análise dinâmica de sistemas hidráulicos, são as suas compressibilidade e viscosidade, muito embora existam outros factores importantes como o grau de contaminação ou a sua densidade. A compressibilidade e a viscosidade são influenciadas por variações de pressão e de temperatura; no entanto, essas últimas podem ser limitadas através de permutadores de calor instalados no circuito hidráulico, na maioria dos casos no reservatório.

#### 4.1.1 Massa específica volúmica

A massa específica de um fluido define-se como a massa de fluido contida na unidade de volume:

$$r = \frac{DM}{DV} \quad (4.1)$$

Os valores típicos para a massa específica variam entre  $890\text{Kg.m}^{-3}$ , para óleos minerais, e  $1400\text{Kg.m}^{-3}$  para clorohidrocarbonetos. A massa específica de um fluido,  $r$ , varia em pequena proporção com a pressão devido à compressibilidade do fluido. Essa variação pode não ser desprezável devido às variações de pressão elevadas existentes nos sistemas hidráulicos. Segundo

Merrit [Merrit67] os primeiros dois termos da série de Taylor são uma boa aproximação para a descrição da variação da massa específica,  $\rho$ , em função da pressão,  $P$ , e temperatura,  $T$ , do fluido. Então, a equação de estado linearizada de um fluido, válida na vizinhança de  $(r_0, P_0, T_0)$ , é definida por:

$$\rho(P, T) = r_0 + \frac{\partial \rho}{\partial P} (P - P_0) + \frac{\partial \rho}{\partial T} (T - T_0) \quad (4.2)$$

onde  $\rho_0$  é a massa específica à temperatura  $T_0$  e pressão  $P_0$ .

A Figura 4.1 mostra a variação da massa específica, para um óleo hidráulico comercial, com a temperatura e com a pressão. As curvas foram obtidas através da equação (4.2).

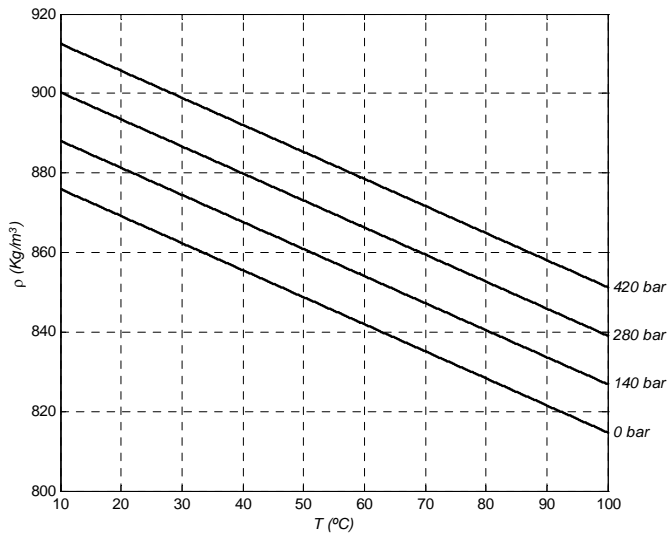


Figura 4.1 Variação de  $\rho$  para o óleo Tellus 68<sup>®</sup> de acordo com (4.2) para  $\frac{\partial \rho}{\partial P} = 8.72 \cdot 10^{-2} \text{Kg m}^{-3} \text{bar}^{-1}$ ,

$$\frac{\partial \rho}{\partial T} = -0.68 \text{Kg m}^{-3} \text{ } ^\circ\text{C}^{-1}, T_0 = 15^\circ\text{C}, P_0 = 0 \text{bar}, r_0 = 872 \text{Kg m}^{-3}$$

## 4.1.2 Módulo de compressibilidade

Quando comprimido numa câmara de um cilindro, o óleo comporta-se como uma mola, introduzindo um sistema massa-mola de segunda ordem, cuja frequência natural limita abruptamente a largura de banda de qualquer servomecanismo hidráulico [Viersma80]. A compressibilidade de um fluido hidráulico contido num volume  $V_0$  e sob uma pressão  $P$  é definida através da expressão:

$$\frac{DV}{V_0} = - \frac{DP}{b_s} \quad (4.3)$$

onde  $\beta_s$  é o chamado módulo de compressibilidade secante e tem a dimensão ( $Nm^{-2}$ ) à semelhança do módulo de Young nos sólidos. O módulo de compressibilidade tangente é definido por:

$$b_t = -V_0 \frac{\partial P}{\partial V} \quad (4.4)$$

A partir de (4.4) é usual definir-se o módulo de compressibilidade isotérmico,  $b$ , e o módulo de compressibilidade adiabático,  $b_a$ :

$$b = -V_0 \left( \frac{\partial P}{\partial V} \right)_T \quad (4.5)$$

$$b_a = -V_0 \left( \frac{\partial P}{\partial V} \right)_S \quad (4.6)$$

onde  $S$  significa entropia.

O módulo de compressibilidade isotérmico, (4.5), pode também exprimir-se em função da massa específica. Assim, à pressão  $P_0$  e temperatura  $T_0$ ,  $\beta$  vem dado por:

$$b = r_0 \times \left( \frac{\partial P}{\partial r} \right)_{T_0} \quad (4.7)$$

No caso de o fluido ser um líquido, como é o caso em hidráulica, segundo Merrit [Merrit67] e Burrows [Burrows72], as pequenas diferenças existentes entre as várias definições para o módulo de compressibilidade, apresentadas anteriormente, são normalmente ignoradas.

Em [Hayhard70] são propostas duas expressões para a variação do módulo de compressibilidade secante adiabático com a temperatura,  $T$ , e com a pressão,  $P$ .

$$b_s = (1.3 + 0.15 \times \log(h)) \times 10^{0.0023 \times (20 - T)} \times 10^3 + 5.6 \times P \text{ (bar)} \quad (4.8)$$

$$b_s = (1.51 + 7 \times (r - 0.86)) \times 10^{0.0023 \times (20 - T)} \times 10^3 + 5.6 \times P \text{ (bar)} \quad (4.9)$$

onde  $h$  (cSt) e  $r$  (Kg/litro) são, respectivamente, a viscosidade cinemática e a massa específica à pressão atmosférica e a 20°C.

O módulo de compressibilidade tangente adiabático é dado por:

$$b_a = b_s \frac{b_s - P}{b_0} \text{ (bar)} \quad (4.10)$$

onde  $b_0$  é o módulo de compressibilidade isotérmico à pressão atmosférica.

O módulo de compressibilidade adiabático relaciona-se com o módulo de compressibilidade isotérmico pela expressão [Merritt67]:

$$b = b_a \frac{C_v}{C_p} \quad (4.11)$$

onde  $C_v$  é o calor específico a volume constante e  $C_p$  é o calor específico a pressão constante.

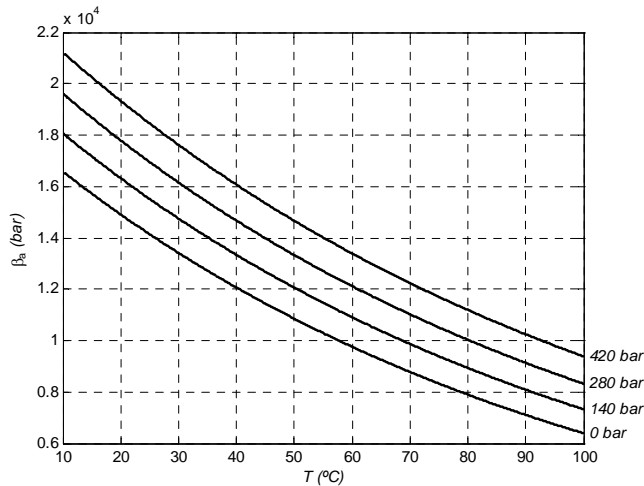


Figura 4.2 Variação do módulo de compressibilidade adiabático usando (4.9) e (4.10) para  $r(20^\circ C) = 0.865cSt$  e  $b_0 = 1.6 \cdot 10^4 bar$

O coeficiente de expansão cúbica,  $\alpha$ , pode também ser definido considerando um volume inicial,  $V_0$ , e a sua variação com a temperatura,  $T$ , a uma dada pressão,  $P$ :

$$a = \frac{1}{V_0} \left( \frac{\partial V}{\partial T} \right)_P \quad (4.12)$$

Se se considerar a massa específica,  $\alpha$  pode definir-se, para o fluido à pressão  $P_0$ , através da expressão:

$$a = - \frac{1}{r_0} \left( \frac{\partial r}{\partial T} \right)_{P_0} \quad (4.13)$$

onde  $\rho_0$  é, como já referido anteriormente, a massa específica à temperatura  $T_0$  e pressão  $P_0$ .

A equação de estado linearizada de um fluido, (4.2), pode então ser definida em função de  $\alpha$  e  $\beta$ :

$$r(P, T) = r_0 \left[ 1 + \frac{1}{b} (P - P_0) - a (T - T_0) \right] \quad (4.14)$$

O módulo de compressibilidade dos fluidos hidráulicos aumenta ligeiramente com a pressão mas diminui fortemente com o aumento da temperatura.

## Módulo de compressibilidade efectivo

O efeito da rigidez de um líquido, caracterizada normalmente pelo seu módulo de compressibilidade, pode ser substancialmente diminuído pela existência de gás e pela elasticidade estrutural dos componentes mecânicos que contêm o líquido. Quando o volume de ar é muito menor que o volume de líquido (o que acontece normalmente nos sistemas hidráulicos) o módulo de compressibilidade efectivo pode ser calculado pela seguinte fórmula [Merritt67]:

$$\frac{1}{b_e} = \frac{1}{b_c} + \frac{1}{b_l} + \frac{1}{b_g} \frac{V_g}{V_t} \frac{\beta_g}{\beta_l} \quad (4.15)$$

onde  $V_g$  é o volume de gás,  $V_t$  é o volume total,  $\beta_e$  é o módulo de compressibilidade efectivo,  $\beta_c$  é o módulo de compressibilidade dos componentes mecânicos,  $\beta_l$  é o módulo de compressibilidade do líquido e  $\beta_g$  é o módulo de compressibilidade do gás. Como se pode concluir da fórmula, o menor módulo de compressibilidade limita o módulo de compressibilidade efectivo do sistema. O módulo de compressibilidade isotérmico do gás,  $\beta_g$ , é igual ao nível de pressão absoluta do gás e, desse modo, uma pequena quantidade de gás no circuito hidráulico pode provocar uma diminuição significativa no módulo de compressibilidade efectivo. As tubagens de ligação entre componentes hidráulicos são as maiores fontes de elasticidade estrutural mecânica. Na maioria dos casos a medida directa é o melhor meio de estimar o módulo de compressibilidade efectivo.

Nos sistemas hidráulicos modernos, o fluido hidráulico é mantido dentro de uma gama estreita de temperaturas e é comum considerar, para efeitos de simulação, a temperatura constante. Quando os sistemas hidráulicos operam numa larga gama de pressões torna-se importante modelar a variação do módulo de compressibilidade com a pressão; como o módulo de compressibilidade efectivo define a rigidez do sistema, a frequência natural pode variar significativamente; por exemplo, uma variação do módulo de compressibilidade efectivo, de 10000 *bar* para 7000 *bar*, pode provocar uma diminuição de cerca de 15% na frequência natural do sistema hidráulico.

## Modelos para o módulo de compressibilidade efectivo

Em qualquer sistema hidráulico existe sempre uma certa quantidade de ar sob a forma natural ou dissolvido no óleo. O ar dissolvido não tem influência no módulo de compressibilidade efectivo. Contudo, o ar existente no óleo sob a forma de bolhas reduz significativamente o módulo de compressibilidade efectivo, principalmente para as baixas pressões. A pressão do óleo tem um efeito muito significativo no valor efectivo do módulo de compressibilidade. Uma razão para o efeito da pressão em  $\beta_e$  resulta do facto de a variação da pressão do óleo variar a taxa de ar que entra e que é incorporado no óleo. À medida que a pressão aumenta dissolve-se algum ar no óleo e,

então, a dependência de  $\beta_e$  com a pressão é atenuada. A temperatura do óleo também tem influência em  $\beta_e$  dado que afecta quer a densidade do ar, quer o tamanho das bolhas de ar no óleo, e, por conseguinte, a compressibilidade do óleo. Muito embora o efeito da temperatura do óleo seja importante para situações dinâmicas, este efeito pode ser ignorado quando a temperatura do óleo é aproximadamente constante [Jinghong94]. Se se utilizarem tubagens metálicas, o efeito da sua rigidez em  $\beta_e$  é pequeno, podendo, numa grande parte das aplicações, ser também ignorado.

Alguns dos modelos (modelos 1, 2 e 3), apresentados a seguir, foram propostos por Jinghong em [Jinghong94]. Estes modelos para o módulo de compressibilidade efectivo apresentam diferentes complexidades e são válidos quando são usadas tubagens rígidas, a temperatura do óleo é mantida aproximadamente constante e as condições de bombagem no circuito hidráulico são constantes.

### Modelo 1

Neste modelo considera-se que o módulo de compressibilidade efectivo,  $\beta_e$ , está relacionado somente com a pressão do fluido hidráulico no circuito, com a quantidade de ar existente no fluido e com a compressibilidade do gás:

$$b_e = \frac{b (1 + 10^{-5} P)^{1+1/g}}{b (1 + 10^{-5} P)^{1+1/g} + 10^{-5} \times R \times (1 - c_1 P)^{\frac{2b}{g}} - 10^{-5} - P \frac{\partial}{\partial P}} \quad (4.16)$$

onde:  $c_1 [Pa^{-1}]$  é o coeficiente de variação do volume de bolhas de gás, devido à variação da taxa de entrada de ar e devido à quantidade de ar dissolvido no óleo;  $b [Pa]$  é o módulo de compressibilidade do óleo sem ar;  $g$  é o quociente dos calores específicos para o ar;  $R[\%]$  é o volume de ar existente por unidade de volume de óleo à pressão atmosférica.

### Modelo 2

O modelo representado pela expressão (4.16) pode ser simplificado se se considerar que a redução do volume das bolhas de ar no óleo, devido ao aumento da pressão de zero até  $P$ , é directamente proporcional à pressão do óleo. Ou seja:

$$b_e = \frac{10^5 + P}{A \times P^2 + B \times P + C} \quad (4.17)$$

com  $A [Pa^{-2}]$ ,  $B [Pa^{-1}]$  e  $C$  dados por:

$$\begin{aligned}
 A &= \frac{R}{b}(c_1 + c_2) \\
 B &= \frac{1-R}{b} + R \times \frac{10^5}{b} - \frac{1}{g} \frac{\partial}{\partial \rho}(c_1 + c_2) \\
 C &= \frac{10^5}{b}(1-R) + \frac{R}{g}
 \end{aligned} \tag{4.18}$$

onde  $c_2 [Pa^{-1}]$  é o coeficiente de variação do volume de bolhas de ar devido ao aumento da pressão.

### Modelo 3

Se a variação do volume de entrada de ar no óleo for desprezável, isto é,  $c_1 = c_2 = 0$ , então (4.17) pode ser simplificada de modo a obter uma expressão válida enquanto o nível da pressão no sistema se mantiver em valores elevados:

$$b_e = \frac{10^5 + P}{B \times P + C} \tag{4.19}$$

com

$$\begin{aligned}
 B &= \frac{1-R}{b} \\
 C &= \frac{10^5}{b}(1-R) + \frac{R}{g}
 \end{aligned} \tag{4.20}$$

Quando se considera a entrada de ar no óleo desprezável, isto é  $R = 0$ , o módulo de compressibilidade efectivo é constante e igual ao módulo de compressibilidade do óleo sem ar:

$$b_e = b \tag{4.21}$$

Esta, (4.21), é expressão para o módulo de compressibilidade efectivo mais usada na prática de engenharia. Como foram desprezados muitos factores podem ocorrer erros significativos na análise dinâmica de sistemas hidráulicos.

### Modelo 4

Beater, em [Beater01], apresenta também um conjunto de modelos empíricos para o módulo de compressibilidade efectivo em função da pressão:

$$b_e = 0.5 \times b \times \log_{10} \frac{P}{P_{ref}} + 1 \frac{\partial}{\partial \rho} \tag{4.22}$$

onde  $P_{ref} [Pa]$  é uma pressão de referência e  $b$  é o módulo de compressibilidade a essa pressão.



## Modelo 5

Outro modelo apresentado em [Beater01] utiliza a característica da função exponencial:

$$b_e = b_{\max} (1 - e^{(b_1 + b_2 P)}) \quad (4.23)$$

onde  $b_{\max}$  [Pa] é o módulo de compressibilidade máximo,  $b_1$  e  $b_2$  [Pa<sup>-1</sup>] são parâmetros e têm valores negativos.

A Figura 4.3 apresenta os gráficos do módulo de compressibilidade efectivo em função da pressão para os diferentes modelos apresentados anteriormente. Os parâmetros dos modelos foram estimados de forma a minimizar o erro quadrático médio, em relação aos valores do módulo de compressibilidade efectivo medidos (apresentados na secção 6.2.1.3). A maior diferença entre os modelos é o seu comportamento a baixas pressões. Por exemplo, o modelo 4 (4.22) tem um módulo de compressibilidade que tende para zero com a diminuição da pressão. Os modelos 2 e 3 são, de entre os vários modelos apresentados e, por comparação com os resultados obtidos experimentalmente, aqueles que apresentam a melhor relação desempenho/complexidade; note-se que somente são necessárias multiplicações e divisões para o cálculo de  $\beta_e$ , enquanto que, para os outros modelos, são necessários cálculos de potências não inteiras, de funções logarítmicas ou de funções exponenciais.

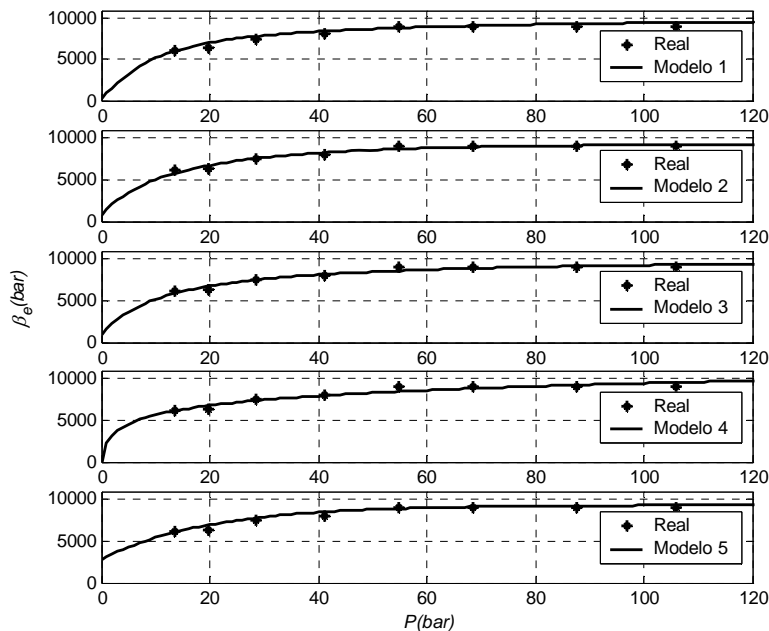


Figura 4.3 Módulo de compressibilidade efectivo em função da pressão para os modelos definidos em (4.16), (4.17), (4.19), (4.22) e (4.23)

### 4.1.3 Viscosidade

Além da tarefa essencial de transmissão de potência, os óleos hidráulicos têm um papel bastante importante na lubrificação dos componentes. A lubrificação consiste em interpor líquido entre duas superfícies tendencialmente em contacto, com o objectivo de evitar o desgaste dos materiais e minimizar as forças de atrito. A lubrificação hidrodinâmica resulta na separação completa das superfícies com movimento relativo. Nestas condições, as únicas forças resistentes a esse movimento relativo são as necessárias para vencer as forças de corte do fluido. A lubrificação hidrodinâmica depende, em grande parte, da viscosidade do fluido. Se a viscosidade for muito baixa, aumentam as fugas, se a viscosidade for demasiado elevada, diminui a eficiência dos componentes com a perda de potência devido ao atrito viscoso.

A velocidade do fluido varia proporcionalmente com a distância entre as superfícies em movimento (escoamento laminar). O facto de existir movimento relativo entre as camadas de fluido significa que existem tensões de corte entre as camadas de fluido. A hipótese de Newton refere que a tensão de corte entre placas de fluido adjacentes, de dimensão infinitesimal, é proporcional à taxa de variação da velocidade na direcção perpendicular ao movimento. Desse modo, a tensão de corte,  $t$ , ao longo da distância infinitesimal  $dy$  é dada por:

$$t = m \frac{dv}{dy} \quad (4.24)$$

onde  $m$  [Pa  $\times$  s] é a viscosidade absoluta ou viscosidade dinâmica.

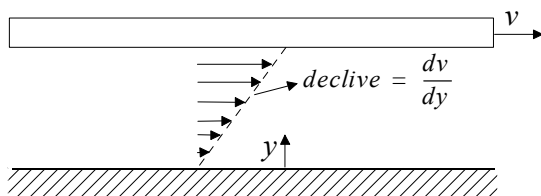


Figura 4.4 Gradiente de velocidade no fluido entre uma placa estacionária e uma placa paralela em movimento

A viscosidade dinâmica é uma medida da tensão de corte entre duas placas paralelas com movimento relativo e quantifica, desse modo, a resistência interna de um fluido ao seu escoamento. A maioria dos óleos hidráulicos são fluidos Newtonianos onde a viscosidade dinâmica é independente das forças de corte.

A razão entre a viscosidade dinâmica,  $m$ , e a massa específica,  $r$ , ocorre em muitas equações e é facilmente medida em viscosímetros. A esta relação chama-se viscosidade cinemática,  $h$  [ $m^2 s^{-1}$ ], sendo dada por:

$$h = \frac{m}{r} \quad (4.25)$$

É comum explicitar-se a viscosidade cinemática em ‘centistoke’:

$$1 \text{ cSt} = 10^{-6} \text{ m}^2 \text{ s}^{-1}$$

Os valores típicos, para os óleos hidráulicos, variam de 5 a 500 cSt e são muito sensíveis a variações de temperatura.

A viscosidade dos líquidos decresce acentuadamente com o aumento da temperatura e aumenta, mais lentamente, com o aumento de pressão. A variação da viscosidade com a temperatura é a mais importante e pode ser aproximada, segundo Blackburn [Blackburn60], pela equação:

$$m = m_0 e^{-l_1(T - T_0)} \quad (4.26)$$

onde  $\mu_0$  é a viscosidade absoluta a uma temperatura de referência  $T_0$ ,  $l_1$  [ $^{\circ}\text{C}^{-1}$ ] é uma constante que depende do líquido e  $T$  é a temperatura.

A equação de Walther [McCabe01] também é normalmente usada para relacionar a variação da viscosidade cinemática com a temperatura. Esta pode ser representada em cSt, em função da temperatura,  $T$  [ $^{\circ}\text{C}$ ]:

$$h = -0.7 + 10^{10(a + b \log T)} \quad (4.27)$$

onde  $a$  e  $b$  são constantes e  $T$  é a temperatura absoluta do óleo. Para o óleo Tellus68,  $a = 9.156$  e  $b = -3.564$  [Silva01].

A variação da viscosidade dos fluidos hidráulicos com a pressão é normalmente desprezada em sistemas hidráulicos que funcionam a baixas pressões. Contudo, para sistemas de altas pressões, essa variação da viscosidade pode ser significativa.

Segundo Khaimovich [Khaimovich65] e Schmidt [Schmidt00], a influência da pressão na viscosidade pode ser representada por:

$$h = h_0 e^{l \cdot P} \quad (4.28)$$

onde  $h_0$  é a viscosidade cinemática, à pressão atmosférica para uma dada temperatura, e  $l$  é o chamado coeficiente viscosidade-pressão. Este coeficiente é também dependente da temperatura e pode ser expresso por:

$$l = c \times T + d \quad (4.29)$$

onde  $c$  e  $d$  são constantes determinadas experimentalmente. Para o óleo hidráulico Tellus68,  $c = -1.1625 \cdot 10^{-5} [^{\circ}\text{C}^{-1}]$  e  $d = 2.6325 \cdot 10^{-3}$  [Silva01].

A Figura 4.5 mostra a evolução da viscosidade cinemática para uma gama de pressões até 300 bar e temperaturas entre 10°C e 100°C. A viscosidade cinemática foi calculada através de (4.28) com  $h_0$  a ser obtido de (4.27). Pode verificar-se a grande dependência da viscosidade com a temperatura.

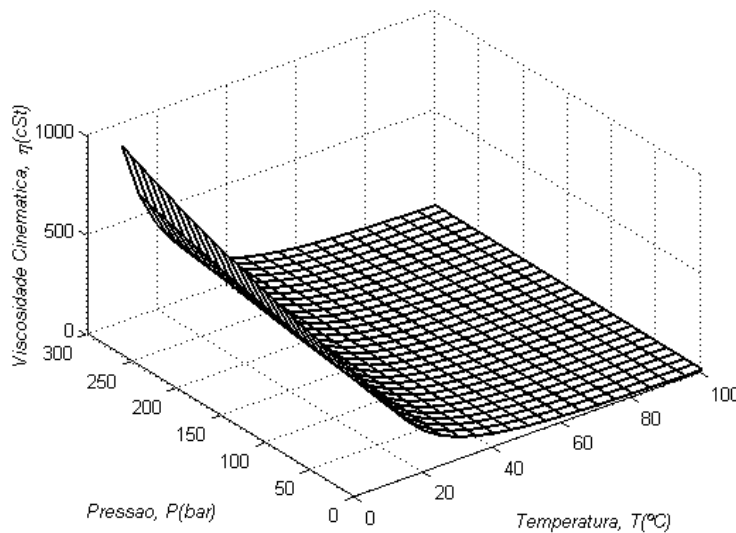


Figura 4.5 Viscosidade cinemática em função da temperatura e da pressão

O método mais utilizado para expressar a relação viscosidade/temperatura é o índice de viscosidade (VI). Quanto mais elevado for o índice de viscosidade, menor será a variação da viscosidade com a temperatura, e vice-versa. A variação da viscosidade com a temperatura é indesejável, pois os componentes hidráulicos são projectados para trabalhar eficientemente numa gama de viscosidades limitada e, idealmente, numa gama de temperaturas alargada.

## 4.2 Modelos de actuadores

Os sistemas hidráulicos são, normalmente, descritos recorrendo à teoria dos fluidos com parâmetros concentrados. Um dos componentes típicos, que pode necessitar de uma descrição através de parâmetros distribuídos, é a linha de transmissão hidráulica. Para a descrição, através de modelos de parâmetros concentrados, de linhas de transmissão hidráulicas é necessário que quaisquer oscilações do sistema sejam significativamente menores do que as causadas pela

propagação das ondas [Watton89]. O comprimento máximo da tubagem que permite uma análise através de parâmetros concentrados é:

$$l < \frac{C_0}{2 \times f} \quad (4.30)$$

onde a velocidade do som no líquido é  $C_0 = \sqrt{\frac{b_e}{\rho}}$ ,  $b_e$  é o módulo de compressibilidade efectivo,  $\rho$  é a massa específica e  $f$  é a máxima frequência de oscilação existente no sistema.

Para a descrição de sistemas hidráulicos, através da teoria dos fluidos com parâmetros concentrados, são usadas duas equações generalizadas [Watton89].

### A equação de continuidade

Considere-se um volume de fluido  $V$ , com um caudal mássico de entrada  $m_i = r_i Q_i$  e um caudal mássico de saída  $m_o = r_o Q_o$  (Figura 4.6)

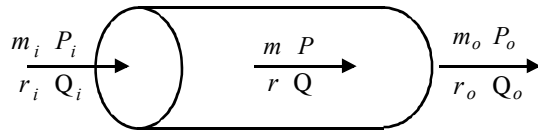


Figura 4.6 Volume de fluido genérico

A continuidade do caudal mássico implica que:

$$r_i Q_i - r_o Q_o = \frac{d}{dt}(rV) \quad (4.31)$$

onde  $Q_i$  é o caudal de entrada e  $Q_o$  é o caudal de saída. Assumindo uma densidade média  $\rho$ , (4.31) pode ser re-escrita:

$$Q_i - Q_o = \frac{dV}{dt} + \frac{V}{r} \frac{dr}{dt} \quad (4.32)$$

Através da equação de estado linearizada (4.14), a versão simplificada do módulo de compressibilidade (4.7) e considerando o módulo de compressibilidade efectivo,  $\beta_e$ , (4.32) pode ser apresentada na sua forma mais usual:

$$Q_i - Q_o = \frac{dV}{dt} + \frac{V}{b_e} \frac{dP}{dt} \quad (4.33)$$

A equação (4.33) é conhecida pela equação de continuidade do escoamento. O primeiro termo do lado direito representa a variação do volume do volume de controlo, devida, por exemplo, ao movimento do êmbolo de um cilindro. O segundo termo representa o caudal devido à compressibilidade efectiva do sistema. Note-se que a deformação das paredes do volume de controlo estão implicitamente inseridas no módulo de compressibilidade efectivo e não devem, por isso, ser contabilizadas no primeiro termo.

Na equação de continuidade (4.33), o módulo de compressibilidade efectivo é suposto constante. Contudo em algumas situações considera-se a sua dependência com a pressão. Interessa então averiguar a sua validade para um módulo de compressibilidade dependente da pressão.

A densidade do óleo depende da temperatura e da pressão, equação (4.2), isto é  $r = r(T, P)$ . Deste modo pode-se estabelecer a seguinte igualdade:

$$dr = \frac{\partial r}{\partial T} dT + \frac{\partial r}{\partial P} dP \quad (4.34)$$

Assumindo condições isotérmicas tem-se:

$$dr = \frac{\partial r}{\partial P} dP \quad (4.35)$$

A partir de (4.7), o módulo de compressibilidade tangente isotérmico, dependente da pressão, vem definido por:

$$b(P) = r \frac{\partial P}{\partial r} \quad (4.36)$$

Usando  $\frac{r}{b(P)} = \frac{\partial r}{\partial P}$  em (4.35), chega-se a

$$\frac{1}{r} dr = \frac{1}{b(P)} dP \quad (4.37)$$

Sendo  $\rho_0$  a densidade do óleo à pressão  $P_0$ , obtém-se a seguinte expressão para  $\rho$  à pressão  $P$ :

$$r = r_0 e^{\int_{P_0}^P \frac{1}{b(P)} dP} \quad (4.38)$$

A variação no tempo de  $\rho$  pode ser calculada por  $\frac{dr}{dt} = \frac{dr}{dP} \frac{dP}{dt}$ , em que  $\frac{dr}{dP}$  pode ser calculado através de:

$$\frac{dr}{dP} = \frac{d \left( r_0 e^{\int_{P_0}^P \frac{1}{b(P)} dP} \right) \frac{\ddot{\theta}}{\theta}}{dP} = r_0 e^{\int_{P_0}^P \frac{1}{b(P)} dP} \frac{d \left( \frac{r}{r_0} \frac{1}{b(P)} \right) \frac{\ddot{\theta}}{\theta}}{dP} = r \frac{d \left( \frac{1}{b(P)} \right) \frac{\ddot{\theta}}{\theta}}{dP}. \quad (4.39)$$

Mas

$$\frac{d \left( \frac{r}{r_0} \frac{1}{b(P)} \right) \frac{\ddot{\theta}}{\theta}}{dP} = \frac{d \left( \text{Pr} \int \frac{1}{b(P)} \frac{\ddot{\theta}}{\theta} - \text{Pr} \int \frac{1}{b(P)} \frac{\ddot{\theta}}{\theta} \right)}{dP} \quad (4.40)$$

onde  $\text{Pr} \int \frac{1}{b(P)} \frac{\ddot{\theta}}{\theta}$  é a primitiva de  $\frac{1}{b(P)}$  à pressão  $P$ . Como  $\text{Pr} \int \frac{1}{b(P)} \frac{\ddot{\theta}}{\theta}$  é um valor constante

(4.40) pode ser simplificada:

$$\frac{d \left( \frac{r}{r_0} \frac{1}{b(P)} \right) \frac{\ddot{\theta}}{\theta}}{dP} = \frac{1}{b(P)} \quad (4.41)$$

e então  $\frac{dr}{dP} = r \frac{1}{b(P)}$ .

A equação de continuidade do escoamento (4.32) pode, assim, ser re-escrita para um módulo de compressibilidade mesmo que dependente da pressão como:

$$Q_i - Q_o = \frac{dV}{dt} + \frac{V}{b(P)} \frac{dP}{dt} \quad (4.42)$$

### A equação generalizada da quantidade de movimento

A aplicação da 2ª lei de Newton ao movimento do volume de fluido, Figura 4.6, e assumindo que a massa do fluido no volume de controlo,  $M$ , é constante, resulta em

$$(P_i A_i - P_o A_o) - \hat{\Sigma} F_r = M \frac{dv}{dt} \quad (4.43)$$

onde  $\hat{\Sigma} F_r$  é o somatório de todas as forças resistentes ao movimento do fluido, que podem ser, por exemplo, forças de atrito viscoso.  $A_i$  e  $A_o$  são, respectivamente, a secção de entrada e de saída do volume de controlo,  $M$  é a massa de fluido existente no volume de controlo e  $v$  é a velocidade média do fluido.

## Modelo de actuador linear assimétrico (cilindro de duplo efeito)

Considere-se o diagrama de um actuador linear assimétrico apresentado na Figura 4.7.

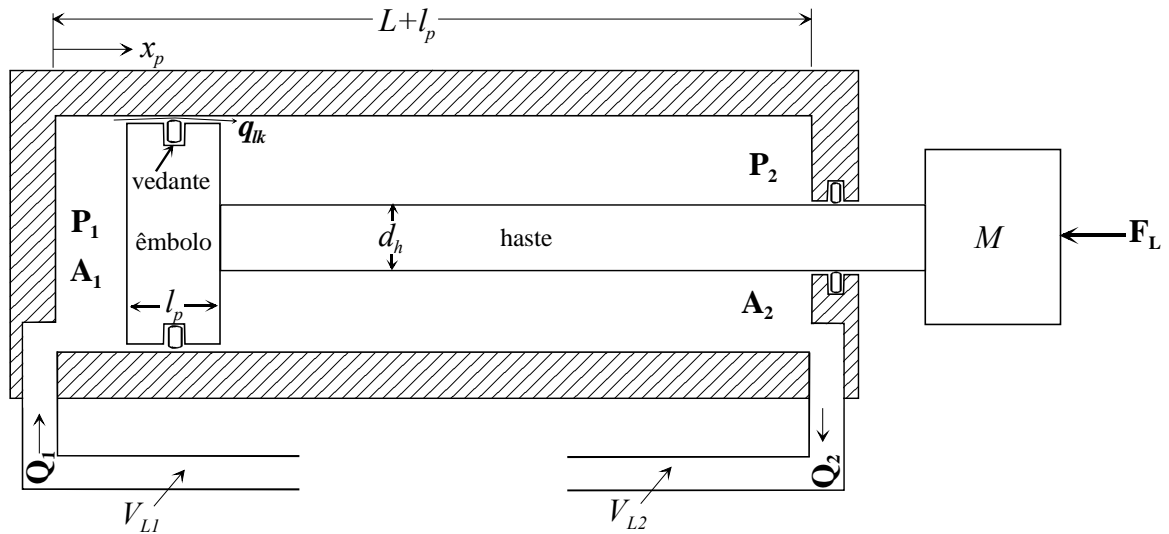


Figura 4.7 Actuador linear assimétrico (cilindro de duplo efeito)

Os caudais envolvidos na movimentação do actuador podem ser estabelecidos pela aplicação de (4.42) a cada uma das câmaras do cilindro. Os volumes das tubagens de alimentação,  $V_{L1}$  e  $V_{L2}$ , são contabilizados nas respectivas câmaras do cilindro, de tal forma que os volumes em compressão numa determinada posição,  $x_p$ , do êmbolo, são  $V_1 = V_{L1} + A_1 x_p$  e  $V_2 = V_{L2} + A_2 (L - x_p)$  para as câmaras 1 e 2 do cilindro, respectivamente.

O caudal de entrada na câmara 1 é dado por:

$$Q_1 = g_{lk} (P_1 - P_2) + A_1 \times v_p + \frac{V_{L1} + A_1 x_p}{\beta_{e1} (P_1)} \frac{dP_1}{dt} \quad (4.44)$$

onde o termo  $q_{lk} = g_{lk} (P_1 - P_2)$  representa o caudal de fugas internas do cilindro (consideram-se as fugas externas nulas). Segundo Merrit [Merrit67], o caudal de fugas pode assumir-se laminar e é quantificado pela condutância de fugas,  $g_{lk}$ . O caudal de saída da câmara 2 é dado por:

$$Q_2 = g_{lk} (P_1 - P_2) + A_2 \times v_p - \frac{V_{L2} + A_2 (L - x_p)}{\beta_{e2} (P_2)} \frac{dP_2}{dt} \quad (4.45)$$

$\beta_{e1}$  e  $\beta_{e2}$  são os módulos de compressibilidade efectivos referentes às duas câmaras do cilindro e representam a compressibilidade do óleo, tubagens e paredes do cilindro. Note-se que  $\beta_{e1}$  e  $\beta_{e2}$  podem ser dependentes da pressão nas câmaras do cilindro. A velocidade do êmbolo é dada por

$$v_p = \frac{dx_p}{dt} \quad (4.46)$$



A dinâmica do êmbolo do cilindro é definida por:

$$M \times a_p = P_1 A_1 - P_2 A_2 - F_a - F_L \quad (4.47)$$

onde a aceleração do êmbolo é:

$$a_p = \frac{dv_p}{dt} \quad (4.48)$$

$F_L$  é a força da carga,  $F_a$  é a força de atrito e  $M$  é a massa total em movimento (carga, êmbolo, haste e massa de óleo no interior do cilindro).

A utilização de um modelo para o cilindro em qualquer situação, nomeadamente simulação em tempo real onde todos os modos de operação podem ocorrer, implica que os limites para o movimento do êmbolo estejam perfeitamente definidos (neste caso  $0 \leq x_p \leq L$ ). A modelação do comportamento do actuador hidráulico quando o êmbolo atinge um dos extremos do cilindro é, por isso, fundamental para o uso do modelo em todas as situações. Podem-se encontrar diferentes modelos na literatura, ([Brach91], [Brogliato96], [Bullo98] ou [Chatterjee98]), para modelação do impacto ou colisão em corpos rígidos. Esses modelos caracterizam-se pela modelação da força elástica, do amortecimento e do coeficiente de restituição. Contudo, a dinâmica destes modelos é normalmente muito rápida e, por isso, aumentam a rigidez do sistema de equações, implicando uma diminuição do passo de integração o que impossibilita a simulação em tempo real em plataformas de baixo custo. Para a simulação em tempo real propõe-se a modelação dos limites do actuador através de um modelo híbrido. A forma como o modelo deve ser implementado pode ser descrita através de um statechart híbrido (Figura 4.8).

O statechart possui três sub-estados do tipo OR que correspondem às regiões possíveis de operação para o êmbolo: limite à direita, *StopRight*, limite à esquerda, *StopLeft*, e regime de operação normal, *Normal*. Estes estados podem ser decompostos de modo a obter uma descrição de mais baixo nível do comportamento. Por exemplo, os estados *StopLeft* e *StopRight* são decompostos de modo a descrever o comportamento do cilindro quando está parado nos limites físicos. A principal acção para suportar os limites rígidos do actuador é a re-inicialização da variável de estado,  $v_p$ , quando o êmbolo atinge um dos limites ( $x_p < 0$  ou  $x_p > L$ ). Quando ocorre este evento o actuador “entra” num estado de paragem atribuindo-se o valor nulo à aceleração do êmbolo ( $a_p = 0$ ) e inicializando-se velocidade ( $v_p := 0$ ). Para “abandonar” este estado de paragem, a força aplicada resultante (força hidráulica menos a força de atrito e força da carga) deve inverter o seu sinal. As actividades *During* definem o comportamento contínuo enquanto o statechart permanecer com os estados activos. O estado *Normal* especifica que a aceleração do êmbolo,  $a_p$ , é definida usando a Segunda lei de Newton, onde  $f$  é a força efectiva aplicada ao êmbolo e  $M$  é a massa total em

movimento. Na activação dos estados *StopLeft* ou *StopRight*, a acção associada à entrada no estado re-inicializa a velocidade do êmbolo. Ao mesmo tempo, considerando o caso da entrada em *StopRight*, o actuador passa ao estado *PosForceR* (sub-estado *default* de *StopRight*) e activa a sua actividade *During*, a qual define a aceleração do êmbolo como nula. Se, entretanto, a força aplicada resultante, definida na actividade *During* do estado *CylinderWithLimits*, inverter o seu sinal, a aceleração é novamente regulada pela segunda lei de Newton, já que o estado *NegForceR* é activado. Note-se que todas as actividades *During* dos estados activos definem, num determinado momento, o sistema de equações diferenciais e algébricas a ser resolvido para a simulação do comportamento do cilindro nesse seu modo contínuo.

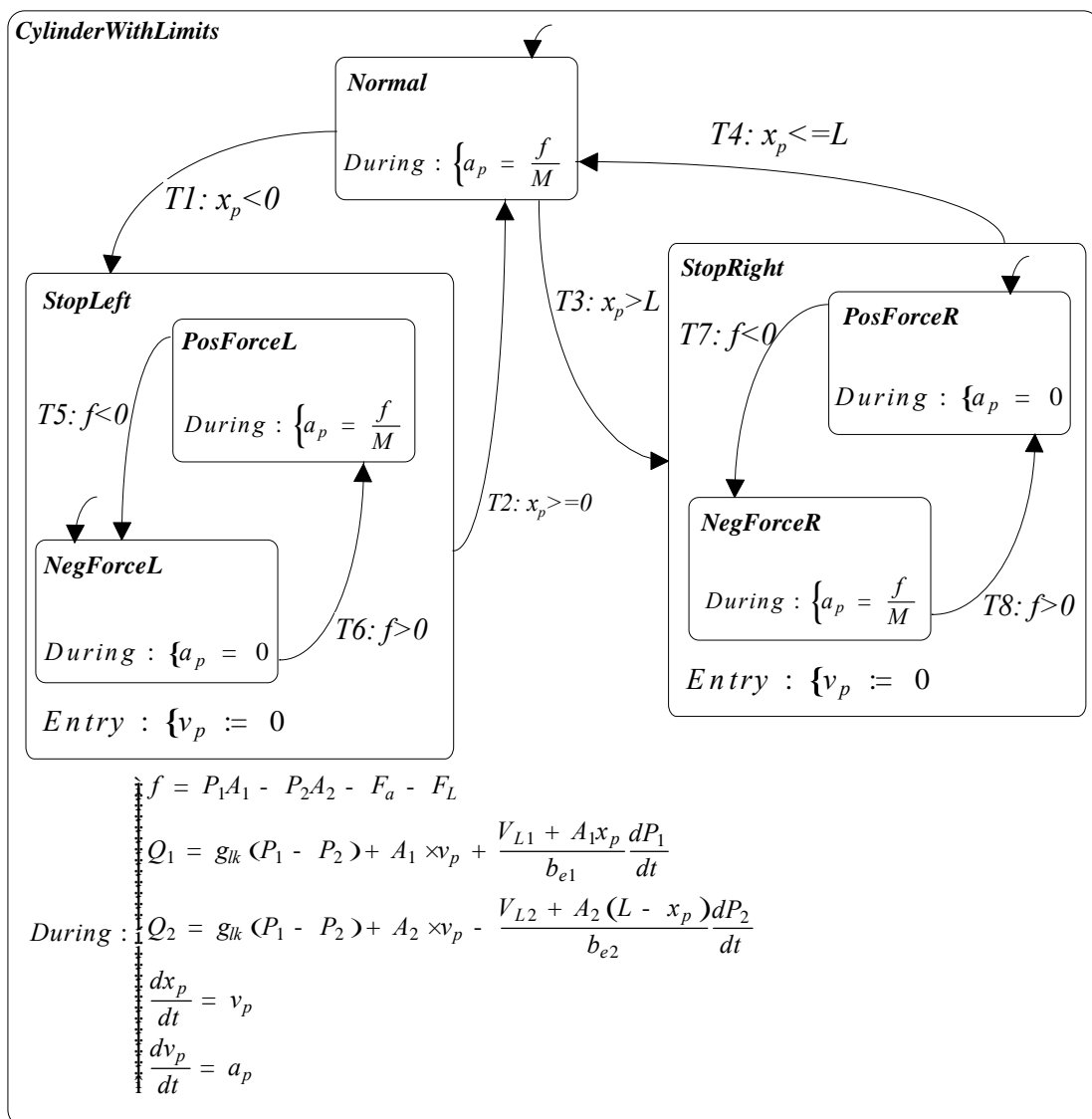


Figura 4.8 Statechart híbrido para descrição do comportamento de um actuador hidráulico linear assimétrico com suporte para limites rígidos.

### 4.3 Modelação do atrito

O atrito é um fenómeno altamente não linear e pode provocar erros em regime estacionário e baixo desempenho em sistemas dinâmicos. O termo atrito é normalmente usado para designar uma força tangencial de reacção, entre duas superfícies em contacto, responsável pela dissipação de energia. De uma forma mais precisa, pode definir-se o atrito como a força necessária para a deformação elástica e plástica das rugosidades em contacto, quando existe movimento relativo entre duas superfícies. Este fenómeno pode ser usado como um efeito positivo que permite abrandar ou parar um determinado movimento (por exemplo, numa acção de travagem), ou pode representar um efeito indesejável que provoca aquecimento dos materiais em contacto e perturbações no movimento. Podem considerar-se vários fenómenos de atrito tendo em conta as superfícies em contacto. Por exemplo, entre superfícies lisas com contactos secos, o atrito pode ser modelado através de forças de deformação elásticas e/ou plásticas das rugosidades em contacto. No caso de contactos lubrificados, como é geralmente o caso do fenómeno do atrito em componentes hidráulicos, estão envolvidos outros mecanismos físicos no atrito. Para baixas velocidades, o fluido actua como uma camada superficial fina e as forças de corte determinam as forças de atrito. Para velocidades altas desenvolve-se uma camada de fluido na superfície devido a efeitos hidrodinâmicos. Estas forças de corte dependem da viscosidade do fluido, assim como da distribuição de velocidades na camada de fluido. Quando, para altas velocidades, as pressões envolvidas são elevadas o fluido entre os contactos é transformado numa fase sólida amorfa, devido à pressão elevada [Olsson96].

O coeficiente de atrito, em sistemas lubrificados, normalmente decresce quando a velocidade aumenta a partir do zero. Quando a espessura da película de fluido é suficientemente grande para separar completamente os dois corpos em contacto, o coeficiente de atrito pode aumentar com a velocidade, pois os efeitos hidrodinâmicos tornam-se significativos. A espessura da película de fluido é um parâmetro muito importante na análise do atrito lubrificado. Os mecanismos que levam à formação dessa película incluem a dinâmica do sistema. Estes pressupostos levaram ao desenvolvimento de modelos dinâmicos para o atrito, tendo, nos últimos anos, aumentado o interesse na dinâmica do atrito, principalmente para a análise de servomecanismos de elevada precisão e para o projecto de compensadores de atrito.

### 4.3.1 Atrito em regime de velocidade estacionária (modelos estáticos)

A força de atrito em função da velocidade, para um movimento com velocidade constante, é chamada a curva da Stribeck. Em particular, o decaimento da força para baixas velocidades é denominado o efeito de Stribeck. Os modelos clássicos de atrito utilizam diferentes componentes para representação de cada uma das parcelas do atrito.

#### Atrito de Coulomb + atrito viscoso

O modelo de atrito mais simples modela o atrito de Coulomb e o atrito viscoso:

$$f_a(v) = f_{co} \times \text{sign}(v) + k_v \times v \quad (4.49)$$

onde  $f_{co}$  é o parâmetro de Coulomb,  $k_v$  é o coeficiente de atrito viscoso e  $v$  é a velocidade relativa entre as superfícies em contacto. O atrito de Coulomb representa a componente do atrito proporcional à força normal. O atrito viscoso representa a força de atrito causada pela viscosidade dos lubrificantes.

Uma variante para a componente viscosa do atrito é proposta em [Olsson97] para uma melhor reprodução de dados experimentais:

$$f_a(v) = f_{co} \times \text{sign}(v) + k_v \times |v|^{d_v} \times \text{sign}(v) \quad (4.50)$$

onde  $d_v$  depende da geometria da aplicação.

Devido à sua simplicidade, o modelo (4.49) não permite a descrição do fenómeno do atrito a baixas velocidades porque, nestes casos, ocorre o efeito de Stribeck que é caracterizado por uma diminuição da força de atrito com o aumento da velocidade.

#### Atrito de Coulomb + atrito viscoso + efeito Stribeck

Um dos modelos mais populares que incorpora o efeito de Stribeck é o seguinte:

$$f_a(v) = f_{co} \times \text{sign}(v) + f_v \times v + f_s \times \text{sign}(v) \times e^{-\left(\frac{v}{v_s}\right)^2} \quad (4.51)$$

onde  $f_s = f_{bw} - f_{co}$  é a chamada força de Stribeck,  $f_{bw}$  é o chamado atrito estático (força de *break-away*) e  $v_s$  a velocidade crítica de Stribeck. Esta curva é facilmente obtida através da medição da força de atrito para velocidades constantes.

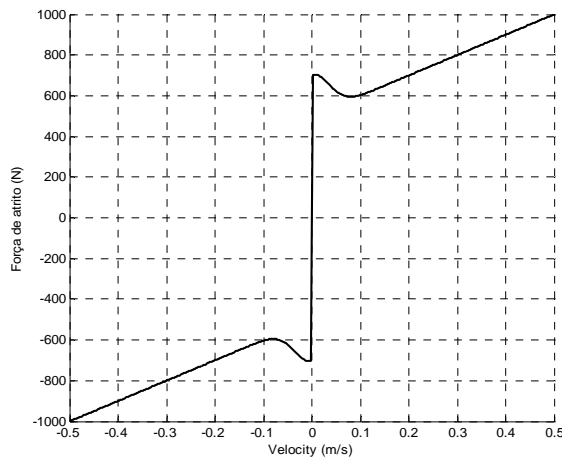


Figura 4.9 Gráfico de (4.51) para  $f_{co} = 500N$ ;  $f_s = 200N$ ;  $v_s = 0.05ms^{-1}$ ;  $k_v = 1000Nms^{-1}$

Uma das implicações do efeito de Stribeck é, em muitos casos, originar um ciclo limite cola-desliza (*stick-slip*) a baixas velocidades.

### O modelo de Beater

Outro modelo para a força de atrito de Coulomb, o efeito de Stribeck e o atrito viscoso é proposto em [Beater98]. Este modela o atrito de forma estática em função da velocidade relativa:

$$f_a = (f_{co} + f_s e^{-c_e |v|}) \text{sign} \langle v \rangle + k_v \cdot v \quad (4.52)$$

onde  $c_e$  é o coeficiente de decaimento da força de Stribeck.

### O modelo de Karnopp

Nenhum dos modelos de atrito apresentados anteriormente especifica a força de atrito para a velocidade zero. De facto, o atrito na paragem (velocidade zero) é uma função da força externa e não da velocidade. Dessa forma a força de atrito pode tomar qualquer valor entre  $-f_{bw}$  e  $f_{bw}$ , o que leva a várias soluções no sistema de equações do movimento, podendo causar problemas numéricos. Estas dificuldades levaram Karnopp [Karnopp85] a propor uma lei para a força de atrito como a apresentada na Figura 4.10, onde o atrito é sempre função da velocidade dos dois corpos em contacto. Karnopp define então uma região de baixa velocidade para  $-v_{\min} < v < v_{\min}$  onde se considera que velocidade é nula. Aqui, a força de atrito é determinada pelas outras forças no sistema, de tal forma que  $v$  permanece na região enquanto a força resultante não ultrapassa a força estática de *break-away*. Fora desta região, a força de atrito é uma função da velocidade. O statechart da Figura 4.11 descreve uma implementação possível para o modelo de Karnopp.

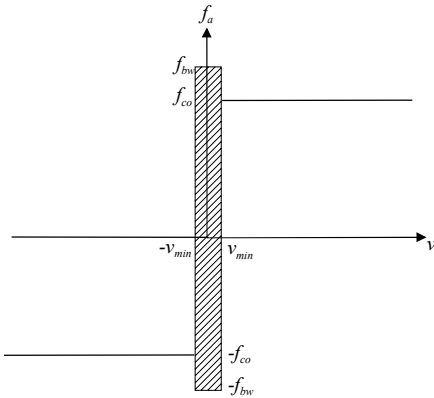


Figura 4.10 Modelo de atrito proposto por Karnopp

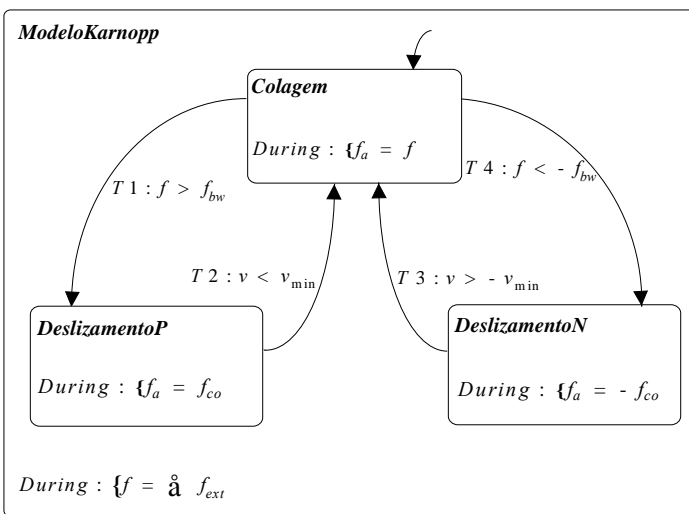


Figura 4.11 Statechart para descrição do comportamento do modelo de Karnopp

Uma das formas mais conhecidas de como o atrito afecta os sistemas de controlo é o fenómeno do movimento cola-desliza, conhecido por *stick-slip*. Este é um problema muito comum, e normalmente indesejável, para movimentos a baixa velocidade. São típicos os movimentos irregulares, com períodos de colagem e deslizamento, que diminuem o desempenho dos sistemas de controlo e podem causar vibrações e ruído. O fenómeno de *stick-slip* deve-se ao facto de a força de atrito, à velocidade nula, ser maior que a baixa velocidade. O exemplo apresentado em [Olsson96], Figura 4.12, onde a força de atrito,  $f_a$ , é modelada pelo statechart da Figura 4.11, gera um movimento *stick-slip*.

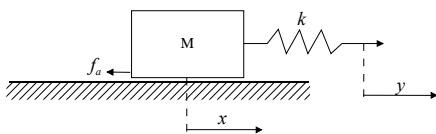


Figura 4.12 Exemplo com movimento *stick-slip*

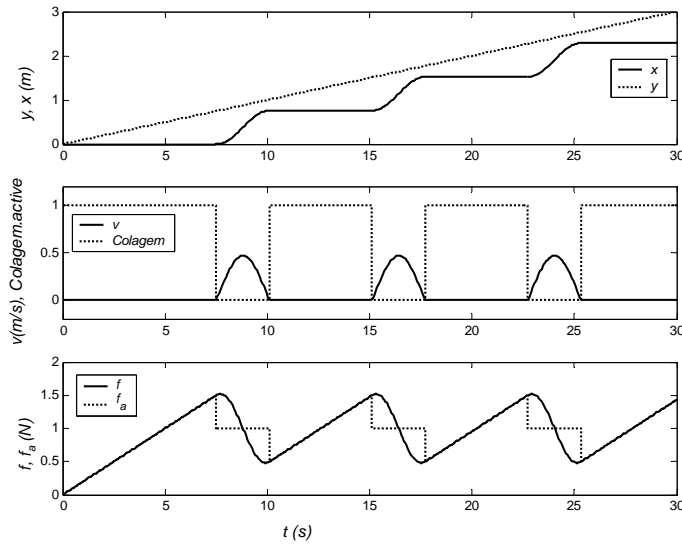


Figura 4.13 Simulação do movimento *stick-slip*

A força externa,  $f = k(y - x)$ , resulta da aplicação de um movimento com velocidade constante,  $v_{ref}$ , no extremo da mola de rigidez  $k$ . As equações do movimento são dadas por:

$$\begin{aligned}
 M \frac{dv}{dt} &= k(y - x) - f_a \\
 \frac{dx}{dt} &= v \\
 \frac{dy}{dt} &= v_{ref}
 \end{aligned}
 \tag{4.53}$$

A Figura 4.13 mostra uma simulação do exemplo ilustrado na Figura 4.12 e em (4.53) com os seguintes parâmetros:  $f_{co} = 1N$ ,  $f_s = 1.5N$ ,  $k = 2Nm^{-1}$  e  $v_{ref} = 0.1ms^{-1}$ . No estado inicial de colagem, a força de atrito equilibra a força externa, que sobe linearmente à medida que  $y$  aumenta, e a massa permanece em repouso. Quando a força externa ultrapassa a força de *break-away*,  $f_{bw}$ , o sistema entra no estado *DeslizamentoP*, a massa começa a deslizar e a força de atrito,  $f_a$ , iguala a força de atrito de Coulomb,  $f_{co}$ . À medida que a massa acelera a força  $f$  diminui (contração da mola) e a massa desacelera, parando de seguida e activando o estado *Colagem*. Este processo é periódico e denomina-se por ciclo limite.

O exemplo seguinte ilustra a oscilação que decorre do controlo PID de posição, na presença de atrito, onde a posição controlada oscila em torno da posição de referência, nunca conseguindo estabilizar (fenómeno designado por *hunting*). Se não existir atrito os parâmetros do controlador  $\{k_p = 2; k_d = 2; k_i = 1\}$ , [Olsson96], fornecem um bom desempenho no controlo de posição de uma massa deslizante, Figura 4.14.

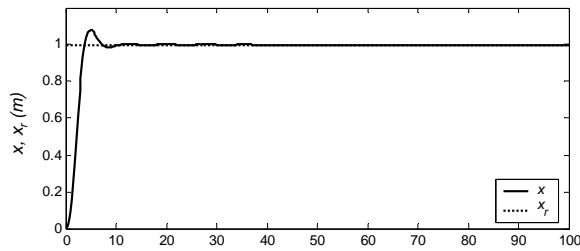


Figura 4.14 Controle PID sem atrito

A introdução de uma força de atrito (usando o modelo de Karnopp implementado através do statechart da Figura 4.11) com  $f_{co} = 0.5N$ ;  $f_s = 1N$ ;  $M = 1Kg$ ;  $x_r = 1m$ , origina um ciclo limite, como se pode ver na Figura 4.15.

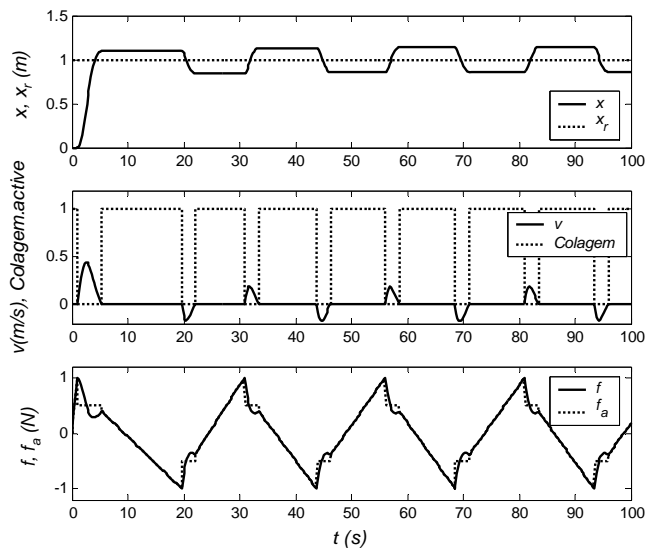


Figura 4.15 Simulação do controle PID de posição com o modelo de atrito de Karnopp

Após um curto estado de colagem a massa movimentar-se rapidamente, passa a posição desejada, abrandar e entrar num estado de *Colagem*. A ação integral diminui então a força  $f$  e o sistema entra no estado *DeslizamentoN*, estabelecendo-se a força de atrito em  $f_{co}$ . Contudo, o movimento ultrapassa novamente a posição desejada, entrando novamente no estado de *Colagem*. O erro de posição determina o tempo necessário à ação integral para atingir um nível de força de *break-away*, e, por conseguinte, o tempo em que o sistema se encontra no estado *Colagem*.

Os modelos apresentados anteriormente consideraram o atrito para velocidades constantes, em regime permanente. Nos modelos seguintes (modelos dinâmicos) considera-se o efeito da variação da velocidade, o qual origina ciclos de histerese cuja dimensão aumenta com a carga normal, com a viscosidade do fluido e com a frequência da variação da velocidade.



## O modelo de Armstrong

O modelo clássico modificado, proposto por Armstrong [Armstrong94], contempla alguns fenómenos de atrito dinâmico, muito embora seja um modelo estático, pois introduz dependências temporais para a colagem e para o efeito de Stribeck. Este modelo é válido quando existe lubrificação e contempla quatro regimes de velocidade, ilustrados na figura Figura 4.16, e duas propriedades dependentes do tempo.

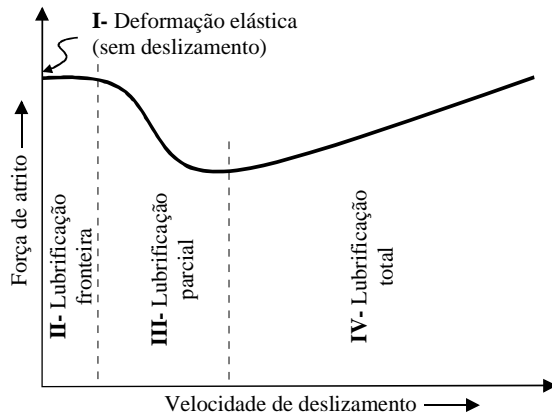


Figura 4.16 Curva de Stribeck com 4 regimes de lubrificação (atrito lubrificado)

De facto, a presença dum lubrificante entre as superfícies em contacto influencia muito as características do deslizamento, pois são criadas camadas finas de lubrificante nas superfícies. A espessura da camada aumenta com a velocidade até espessuras que podem atingir, e mesmo ultrapassar, a altura das rugosidades superficiais.

Estes regimes de velocidade contribuem para a dinâmica do sistema que acelera desde a velocidade nula. A curva define três regimes com movimento e um regime em que a força de atrito não depende da velocidade.

O primeiro regime representa o atrito estático e o deslocamento microscópico que ocorre no pré-deslizamento. Neste regime, quando as forças aplicadas são inferiores ao atrito estático (força de *break-away*), estas resultam em deformações elásticas reversíveis das rugosidades em contacto. As junções entre rugosidades comportam-se linearmente como molas e existe um micro-deslocamento que, por esse facto, pode considerar-se linear com a força aplicada. A força de atrito pode então ser aproximada por:

$$f_a \langle x \rangle = -k_t \times x \quad (4.54)$$

onde  $k_t$  é a rigidez do contacto que depende da geometria das rugosidades, da elasticidade do material e da força normal aplicada;  $x$  é o movimento relativo em relação à posição de equilíbrio, Figura 4.17. Armstrong, em [Armstrong94], apelida esta força como uma “força tangencial”, pois

não é uma força dissipativa nem uma consequência do deslizamento. As curvas tensão-deformação dos materiais definem uma tensão máxima de deformação elástica dum determinado material. Essa tensão máxima define então um limite para o movimento elástico na colagem (2 a  $5\mu\text{m}$  no caso de “junções em aço”, [Armstrong91]). Existe também, contudo, uma deformação plástica das rugosidades, e da camada de fluido lubrificante entre elas, o que dá origem a “pequenos” deslizamentos e a um aumento do atrito estático (*rising friction*). A força de atrito estático é dependente da taxa de aumento da força externa e aumenta com o tempo de permanência num estado de colagem (conhecido como o *dwell time*). Quando a força aplicada excede o atrito estático, as junções estabelecidas entre as rugosidades quebram-se e ocorre então o deslizamento, Figura 4.18.

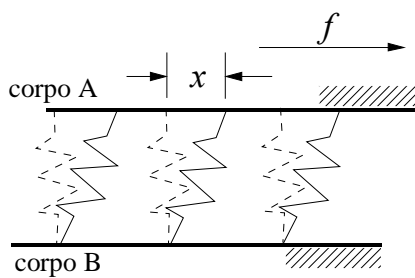


Figura 4.17 Deformação das rugosidades para uma força aplicada inferior ao atrito estático (deslocamento pre-deslizamento),  $f < f_{bw}$

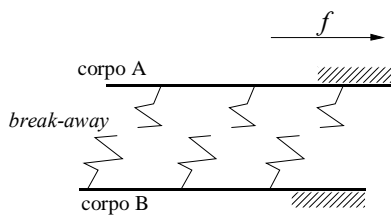


Figura 4.18 Início do deslizamento após o *break-away*,  $f > f_{bw}$

O segundo regime é caracterizado por uma lubrificação de fronteira. Para velocidades muito baixas a lubrificação não é muito importante pois a velocidade não é suficiente para desenvolver uma camada de fluido entre as superfícies.

O terceiro regime é caracterizado por uma lubrificação parcial. À medida que a velocidade aumenta mais lubrificante é deslocado para a interface, e pode vir a formar-se uma camada de fluido. Quanto maior for a viscosidade e a velocidade relativa, tanto mais espessa será a camada de fluido. Se a camada de lubrificante não for suficientemente espessa, algumas cristas das rugosidades podem ainda estar em contacto e, dessa forma, existir uma situação de lubrificação parcial. À medida que a velocidade aumenta, aumenta a espessura da camada de lubrificante diminuindo a

força tangencial resultante, já que as forças ao corte do lubrificante são menores do que as forças ao corte das rugosidades (efeito de Stribeck). Segundo Armstrong este regime é o de mais difícil modelação já que a força de atrito depende da camada de lubrificante (e da sua viscosidade) e também do contacto directo entre algumas das rugosidades. Neste regime observam-se também atrasos temporais entre uma variação da velocidade (ou variações na carga) e a mudança da força de atrito para um novo estado estacionário. Este atraso temporal (ou atraso de fase) é conhecido por memória do atrito (*frictional memory*).

O quarto regime caracteriza-se por uma lubrificação total resultante do facto da camada de lubrificante se manter mais espessa do que o tamanho das rugosidades. As características do atrito são determinadas pela lubrificação hidrodinâmica e elasto-hidrodinâmica. A espessura da camada de lubrificante determina o atrito e depende das características da superfície (rigidez e geometria), da velocidade relativa e da viscosidade do lubrificante.

O modelo proposto por Armstrong contempla também duas propriedades dependentes do tempo:

- Uma relação entre o tempo de permanência num estado de colagem (*dwell time*) e o nível de atrito estático (*rising static friction*);
- Um atraso temporal entre uma variação da velocidade relativa (ou da carga) e a correspondente mudança na força de atrito (*frictional memory*).

O modelo proposto por Armstrong tem dois estados (colagem e deslizante) e sete parâmetros. Na colagem, o deslocamento pré-deslizante define a força de atrito através de:

$$f_a \langle x \rangle = -k_t \times x \quad (4.55)$$

No deslizamento, a força de atrito é definida pelo atrito de Coulomb, pelo atrito viscoso e pelo efeito de Stribeck contemplando o efeito memória de atrito:

$$f_s(v, t) = - \left[ f_{co} + k_v |v| + f_s(g, t_2) \right] \frac{1}{1 + \frac{c}{v_s} \left( \frac{v - t_L}{\phi} \right)^2} \times \text{sign} \langle v \rangle \quad (4.56)$$

com o nível do atrito estático na altura de *break-away* a ser dado por:

$$f_s(g, t_2) = f_{s,a} + (f_{s,\infty} - f_{s,a}) \frac{t_2}{t_2 + g}; \quad (4.57)$$

onde  $f_a$  é a força de atrito instantânea,  $f_{co}$  (\*) é a força de atrito de Coulomb,  $k_v$  (\*) é o coeficiente de atrito viscoso,  $f_s$  (\*) é a amplitude da força de Stribeck,  $f_{s,a}$  é a amplitude da força de Stribeck no fim do período deslizante anterior,  $f_{s,\infty}$  (\*) define a amplitude da força de Stribeck no fim de um

longo período de paragem,  $k_t$  (\*) é a rigidez tangencial do contacto estático,  $v_s$  (\*) é a velocidade crítica de Stribeck,  $t_L$  (\*) é a constante de tempo da memória de atrito,  $g$  (\*) é o parâmetro temporal de estabelecimento do atrito estático,  $t_2$  é o tempo de permanência no estado de colagem (*dwell time*). O símbolo (\*) indica os parâmetros do modelo.

Uma implementação possível para o modelo de Armstrong é ilustrada no statechart da Figura 4.19.

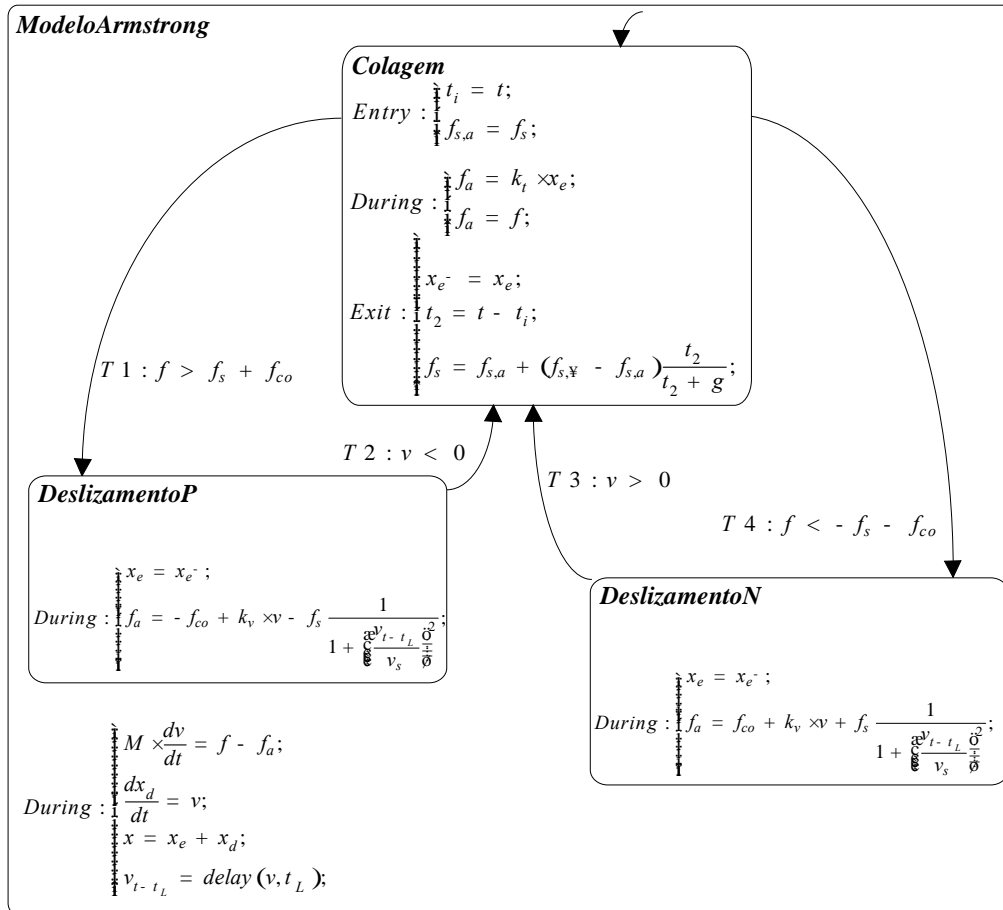


Figura 4.19 Statechart para implementação e teste do modelo de Armstrong

Este descreve um sistema composto por uma massa deslizante,  $M$ , onde é aplicada uma força externa,  $f$ . O movimento da massa deslizante,  $x = x_d + x_e$ , contém uma componente elástica (actuante no pré-deslizamento, constante no deslizamento),  $x_e$ , e uma componente plástica (actuante no deslizamento e constante no pré-deslizamento),  $x_d$ . Na Figura 4.20 interpreta-se o movimento através da concentração de todos os contactos das rugosidades num único “espeto”. A força externa,  $f$ , e o tempo actual,  $t$ , deve ser fornecidos continuamente ao statechart, que devolve ao sistema a força de atrito instantânea,  $f_a$ , a posição da massa, a sua velocidade e o estado do sistema. A velocidade no instante  $t - t_L$ , é designada por  $v_{t-t_L} = v(t - t_L)$  e pode ser calculada através de um operador, *delay*, normalmente fornecido pelas linguagens de modelação. Note-se que no

estado de *Colagem* a força de atrito,  $f_a$ , é igual à força aplicada,  $f$ ; esta situação, em conjunto com a actividade do estado *ModeloArmstrong*, mantém constante o deslocamento plástico,  $x_d$ , (pois a velocidade,  $v$ , anulou-se na transição para o estado *Colagem*), fazendo com que apenas a componente elástica do deslocamento,  $x_e$ , possa variar.

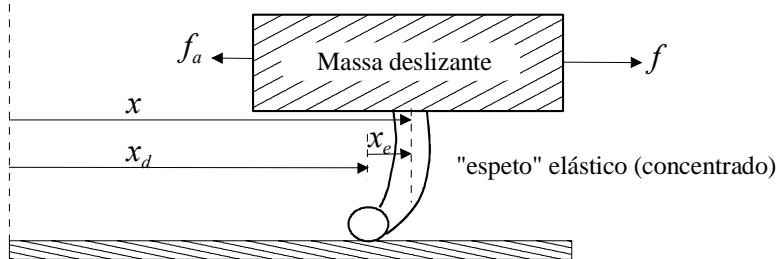


Figura 4.20 Sistema de massa em movimento com aplicação de uma força externa

Em [Armstrong94] é apresentada uma gama de valores para os parâmetros bem como algumas técnicas de parametrização. Lee, em [Lee99], apresenta um processo de identificação dos parâmetros para um modelo semelhante ao modelo de Armstrong. Na simulação do statechart da Figura 4.19 usaram-se os seguintes parâmetros:

$$f_{co} = 100N, f_{s,v} = 50N, g = 100s, k_t = 2 \times 10^5 Nm^{-1},$$

$$v_s = 0.02ms^{-1}, k_v = 1000Nsm^{-1}, t_L = 2 \times 10^{-3}s;$$

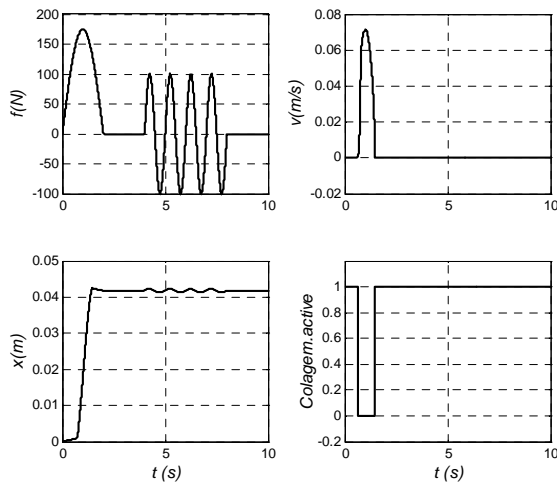


Figura 4.21 Resultados da aplicação de uma força,  $f$ , ao statechart da Figura 4.19

Na Figura 4.21 ilustram-se os resultados da simulação do statechart com a aplicação de uma força, de modo a ilustrar o movimento deslizante e pré-deslizante. De notar que se exagerou propositalmente a amplitude do movimento pré-deslizante para evidenciar a característica elástica

desse movimento. Pode ver-se o comportamento “verdadeiramente” elástico do modelo de atrito no estado de colagem pois, após a anulação da força, a massa regressa à posição que ocupava imediatamente antes de lhe ser aplicada uma força abaixo da força de *break-away*.

### 4.3.2 Modelos dinâmicos

Os modelos dinâmicos de atrito têm particular interesse na concepção de servomecanismos de precisão e permitem implementar compensadores de atrito.

#### O modelo de Dahl

Com o propósito de simular sistemas de controlo com precisão, Dahl desenvolveu um modelo muito simples que é usado principalmente na simulação de sistemas com atrito em rolamentos. O ponto de partida para o modelo de Dahl [Dahl76] é a descrição da curva de tensão-deformação da mecânica dos sólidos clássica, através de uma equação diferencial:

$$\frac{df_a}{dx} = s \left[ 1 - \frac{f_a}{f_{co}} \operatorname{sign} \langle v \rangle \right]^{\frac{\sigma}{\sigma_0}} \quad (4.58)$$

onde  $\sigma$  é o coeficiente de rigidez e  $\alpha$  é um parâmetro que determina a forma da curva tensão-deformação ( $\alpha = 1$  é o valor mais usado).

Uma particularidade do modelo é a sua exclusiva dependência do deslocamento e do sinal da velocidade. No domínio dos tempos o modelo de Dahl pode ser expresso da forma seguinte:

$$\frac{df_a}{dt} = \frac{df_a}{dx} \frac{dx}{dt} = s \left[ 1 - \frac{f_a}{f_{co}} \operatorname{sign} \langle v \rangle \right]^{\frac{\sigma}{\sigma_0}} \times v \quad (4.59)$$

No caso particular de  $\alpha = 1$  o modelo de atrito pode ser descrito por:

$$\frac{df_a}{dt} = s \times v - s \times \frac{f_a}{f_{co}} |v| \quad (4.60)$$

Se em (4.60) se introduzir  $f_a = s \times z$ , o modelo de Dahl pode ser definido por:

$$\begin{aligned} \frac{dz}{dt} &= v - \frac{s \times |v|}{f_{co}} z \\ f_a &= s \times z \end{aligned} \quad (4.61)$$

A Figura 4.22 mostra a força de atrito em função do deslocamento resultante da aplicação de uma força sinusoidal,  $f$ , de amplitude superior à força de atrito de Coulomb,  $f_{co} = 1N$ , a uma massa,  $M = 1Kg$ , e com  $s = 20Nm^{-1}$ .

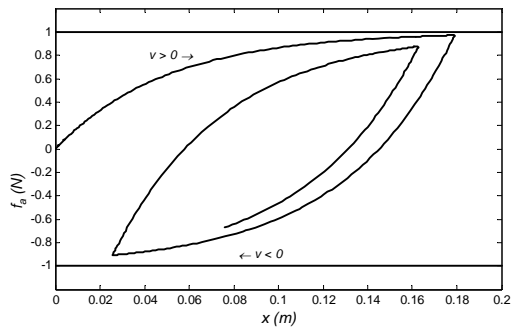


Figura 4.22 Força de atrito,  $f_a$ , em função do deslocamento,  $x$  (modelo de Dahl)

### O modelo de “bristle”

Este modelo considera que, quando duas superfícies estão encostadas, se forma um conjunto de ligações entre pequenos “espetos” pertencentes às duas superfícies [Haessig91]. A distância entre “espetos” nas superfícies é aleatória e tem propriedades estatísticas consistentes com as características macroscópicas do atrito. O modelo considera que o atrito é devido a um grande número de “espetos”, onde cada um contribui com uma pequena parcela para o atrito total. Essa contribuição assume-se proporcional à deformação do “espeto” em cada instante. Quando a deformação de um “espeto” atinge um determinado limite a ligação existente é quebrada e é estabelecida uma nova ligação, mas agora com menor deformação. Pode também considerar-se que o número de ligações estabelecidas, num determinado instante, é uma função da velocidade relativa entre as duas superfícies. Dessa forma a velocidade relativa influencia a força de atrito total através do número de ligações estabelecidas. A força de atrito total é então dada por:

$$f_a = \sum_{i=1}^n \sigma_0 (x_i - b_i) \quad (4.62)$$

onde  $n$  é o número de “espetos” considerados,  $\sigma_0$  é a rigidez dos “espetos”,  $x_i$  é a posição relativa dos “espetos” e  $b_i$  é a localização onde é formada a ligação. A rigidez dos “espetos” pode também ser dependente da velocidade.

A complexidade do modelo aumenta com  $n$ . Haessig, em [Haessig91], refere que podem ser obtidos bons resultados com 20 a 25 “espetos”, sendo também possível, com um único “espeto”, obter comportamentos qualitativos razoáveis. Uma propriedade interessante do modelo é que ele capta a natureza aleatória do atrito. No entanto, o movimento em colagem pode ser oscilatório, pois no modelo não existe amortecimento da deformação dos “espetos”. Devido à sua complexidade, a simulação deste modelo é, em geral, ineficiente. Por estes motivos é proposto no mesmo artigo [Haessig91] um modelo onde o integrador da velocidade de deformação é inicializado.

## O modelo baseado na inicialização do integrador

Este modelo foi desenvolvido de modo a reduzir o tempo de cálculo dispendido pelo modelo dos “espetos” (*bristle*), mas mantendo a capacidade de representar o fenómeno de *stick-slip*. No modelo dos “espetos” as “ligações” entre “espetos” são quebradas e retomadas de uma forma probabilística, daí a necessidade da utilização de vários “espetos”. Neste modelo, a ligação entre dois “espetos” permanece constante através da limitação da deformação no momento da ruptura. O modelo utiliza um estado extra,  $z$ , para determinar a deformação na ligação:

$$\frac{dz}{dt} = \begin{cases} 0 & ;se (v > 0 \wedge z \geq z_0) \text{ ou } (v < 0 \wedge z \leq -z_0) \\ v & ;outros \end{cases} \quad (4.63)$$

A força de atrito é dada por:

$$f_a = (1 + a \langle z \rangle) s_0 \langle v \rangle \times z + s_1 \frac{dz}{dt} \quad (4.64)$$

$s_1 \frac{dz}{dt}$  é o termo de amortecimento que só está activo na colagem, evitando oscilações na região de atrito estático, e  $s_0 \langle v \rangle$  é a rigidez do “espeto” que pode depender da velocidade. O gradiente de colagem,  $a \langle z \rangle$ , funciona como um factor de correcção que só está activo na colagem:

$$a \langle z \rangle = \begin{cases} a_0 & ;se |z| < z_0 \\ 0 & ;outros \end{cases} \quad (4.65)$$

Se  $|z| < z_0$ , o modelo descreve a colagem onde a força de atrito é uma função de  $z$ . Quando a deformação do “espeto” atinge o seu valor máximo  $z_0$ , a força de atrito,  $f_a = (s_0 \langle v \rangle + a_0 s_0 \langle v \rangle) z_0$ , representa a força de *break-away* e  $a_0$  estabelece a diferença entre o atrito estático e o atrito de Coulomb, admitindo que  $s_0 \langle v \rangle$  é constante na colagem. Na descolagem a deformação do “espeto”,  $z$ , mantém-se constante e a força de atrito diminui já que  $a \langle z \rangle$  se anula. No deslizamento, a força de atrito é uma função arbitrária da velocidade dada por  $s_0 \langle v \rangle$ , podendo, nomeadamente, representar o atrito de Coulomb e o atrito viscoso. De facto, num estado de deslizamento, a força de atrito,  $f_a = s_0 \langle v \rangle \times z$ , só depende de  $s_0 \langle v \rangle$ , uma vez que  $z$  permanece constante. O modelo da inicialização do integrador é muito mais eficiente do que o modelo dos “espetos” apresentando, no entanto, descontinuidades em  $z$ . No statechart da Figura 4.23 descreve-se a implementação do modelo de inicialização do integrador.



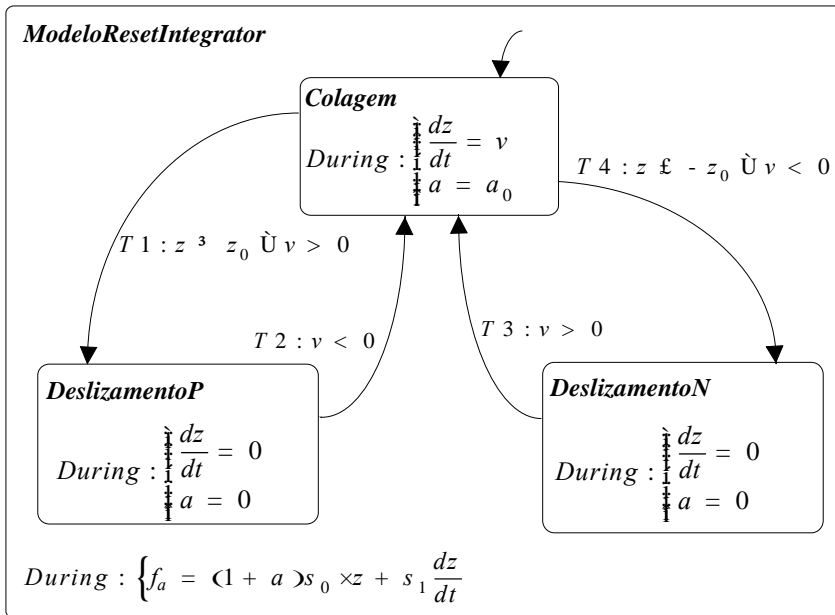


Figura 4.23 Statechart de implementação do modelo de inicialização do integrador

Considerando o exemplo usado para evidenciar o movimento *stick-slip*, Figura 4.12, em conjunto com o modelo descrito no statechart anterior, e utilizando os parâmetros usados em [Haessig91],  $z_0 = 1 \cdot 10^{-5} m$ ,  $s_0 = 2 \cdot 10^4 Nm^{-1}$ ,  $s_1 = 30 Nsm^{-1}$ ,  $a = 0.25$ ,  $k = 100 Nm^{-1}$  e  $M = 0.1 Kg$ , obtêm-se os resultados apresentados na Figura 4.24.

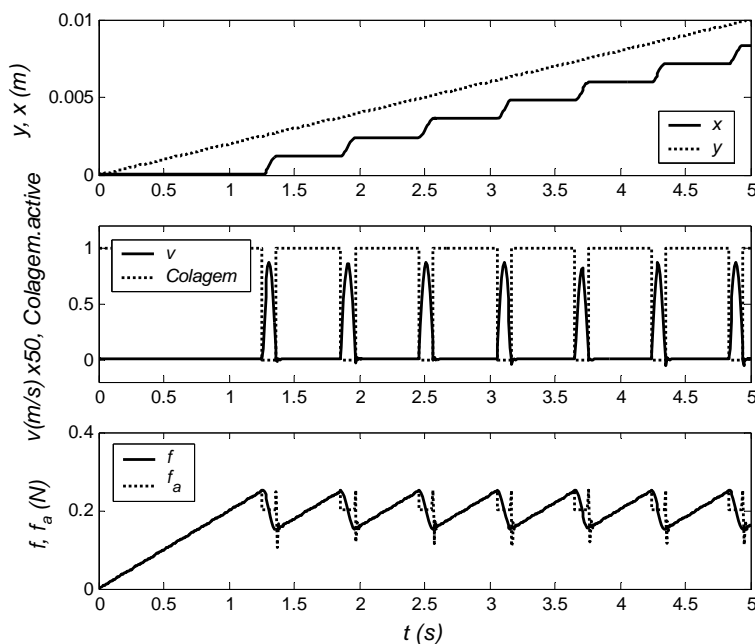


Figura 4.24 Simulação do movimento *stick-slip* para o modelo de inicialização do integrador

## Os modelos de Bliman e Sorine

Os modelos de Bliman e Sorine [Bliman91] têm a sua plataforma matemática assente nos operadores de histerese [Krasnosel89]. Uma análise pormenorizada dos modelos propostos por Bliman e Sorine é apresentada em [Gafvert96]. Bliman e Sorine especificaram inicialmente os modelos através de uma formulação espacial, desenvolvendo de seguida uma formulação temporal para a utilização em simulação e em projecto de compensadores de atrito [Bliman95]. O modelo final inclui a força de colagem, o pré-deslizamento e o comportamento com histerese.

O valor da força de atrito só depende do sinal da velocidade,  $sign \langle v \rangle$ , e da variável espacial,  $e$ , definida da seguinte forma:

$$e = \int_0^t |v \langle t \rangle| dt \quad \text{ou} \quad de = |v \langle t \rangle| dt \quad (4.66)$$

A variável de espaço,  $e$ , é então interpretada como sendo o módulo do deslocamento relativo entre as duas superfícies em contacto, a partir da última mudança no sinal da velocidade. Isto significa que cada mudança de velocidade gera uma nova origem no espaço  $e$ .

Os modelos são expressos como sistemas lineares num formato de espaço de estados:

$$\begin{cases} \dot{x}_e = Ax_e + Bu_e, & x_e(0) = x_0 \in \mathbb{R}^p \\ f_a(x_e) = Cx_e + Du_e \end{cases} \quad (4.67)$$

com:  $A \in \mathbb{R}^{p \times p}, B \in \mathbb{R}^{p \times 1}, C \in \mathbb{R}^{1 \times p}, D \in \mathbb{R}^3$

De entre vários modelos, Bliman e Sorine propõem um modelo de 1ª ordem e um modelo de 2ª ordem. Num modelo de primeira ordem podem ser usados os seguintes valores:

$$A = -\frac{1}{\varepsilon_f}, B = \frac{f_{co}}{\varepsilon_f}, C = 1, D = 0 \quad \text{e} \quad u_e = sign \langle v \rangle$$

onde  $\varepsilon_f$  é uma distância relacionada com o deslocamento no estado de pré-deslizamento.

A especificação deste modelo de 1ª ordem define-se por:

$$\frac{df_a}{dt} = \frac{df_a}{de} \frac{de}{dt} = |v| \frac{df_a}{de} = \frac{f_{co}}{\varepsilon_f} \frac{v}{|v|} - |v| \frac{f_a}{f_{co}} \frac{\dot{v}}{|v|} \quad (4.68)$$

O modelo descrito pela expressão (4.68) é equivalente ao modelo de Dahl (4.59) se  $\sigma = f_{co}/\varepsilon_f$  e  $\alpha = 1$ . O modelo de 1ª ordem não implementa um estado de colagem nem descreve o pico na força de atrito, correspondente à força de *break-away*.

Bliman e Sorine, em [Bliman91], propõe um modelo de 2ª ordem que descreve o deslocamento de pré-deslizamento e a força de *break-away*. Um modelo de segunda ordem pode usar:

$$A = \begin{pmatrix} \frac{1}{\varepsilon_f \eta} & 0 \\ 0 & -\frac{1}{\varepsilon_f h} \end{pmatrix} B = \begin{pmatrix} \frac{f_1}{\varepsilon_f \eta} \\ \frac{f_2}{\varepsilon_f h} \end{pmatrix} C = \begin{pmatrix} 1 \\ 1 \end{pmatrix} D = 0 \quad (4.69)$$

Se se aplicar a transformada de Laplace a (4.67), com as matrizes  $A$ ,  $B$ ,  $C$  e  $D$  dadas por (4.69) e considerando  $u_e = \text{sign} \langle v \rangle$ ,  $y = L \{f_a\}$  e  $u = L \{\text{sign} \langle v \rangle\}$ , tem-se:

$$Y \langle s \rangle = C (sI - A)^{-1} B U \langle s \rangle = \frac{f_1}{\varepsilon_f h s + 1} - \frac{f_2}{\varepsilon_f s + 1} U \langle s \rangle \quad (4.70)$$

$f_1, f_2$  têm a dimensão de força e  $h$  é um parâmetro adimensional.

Através de (4.70) pode constatar-se que o modelo de 2ª ordem é construído a partir de dois modelos de 1ª ordem, um rápido e outro lento, em paralelo [Gafvert96]. O modelo mais rápido descreve uma força de atrito em regime estacionário maior do que o modelo mais lento. Se  $\varepsilon_f$  tender para zero o modelo de 1ª ordem comporta-se como o modelo clássico de atrito de Coulomb, e o modelo de segunda ordem comporta-se como um modelo clássico com atrito de Coulomb e com colagem.

A formulação no domínio dos tempos pode obter-se a partir da formulação espacial (4.67). Com  $de = |v(t)|dt$ ,  $D = 0$ ,  $u_e = \text{sign} \langle v \rangle$  e  $x = x_e$ , tem-se:

$$\begin{cases} \dot{x} = -|v|Ax + Bv \\ y = Cx \end{cases} \quad (4.71)$$

O modelo de 2ª ordem é então:

$$\begin{cases} \dot{x}_1 = -|v| \frac{1}{\varepsilon_f h} x_1 + \frac{f_1}{\varepsilon_f h} v \\ \dot{x}_2 = -|v| \frac{1}{\varepsilon_f} x_2 - \frac{f_2}{\varepsilon_f} v \\ f_a = x_1 + x_2 \end{cases} \quad (4.72)$$

A identificação dos parâmetros do modelo de 2ª ordem resulta da interpretação física do fenómeno do atrito. Para pequenos deslocamentos, menores que o deslocamento de pré-deslizamento, as junções das rugosidades são deformadas elasticamente, dando uma resposta do tipo mola à força exterior (ver Figura 4.28). Para deslocamentos maiores, as rugosidades deformam-se plasticamente tendo a transição entre estas duas fases um comportamento misto. Quando a deformação plástica faz parte do processo, o atrito exhibe uma curva de histerese no gráfico deslocamento versus atrito (ver Figura 4.25). Estas regiões são definidas pelos parâmetros:

$s_e$  – deslocamento de pré-deslizamento

$f_s$  – máxima força de colagem que ocorre em  $s_e$

$s_p$  – deslocamento a partir do qual o comportamento é principalmente plástico

$f_k$  – força de atrito em  $s_p$

$f_s, f_k, s_e, s_p$  são denominados parâmetros de identificação, enquanto que  $f_1, f_2, \varepsilon_f, \eta, k_v$  são chamados parâmetros do modelo, sendo  $k_v$  o coeficiente linear de atrito viscoso.

Uma das formas de identificação experimental dos parâmetros do modelo é através da análise da curva de histerese da força de atrito versus deslocamento num movimento cíclico com mudanças de velocidade. A identificação dos dois pontos  $(s_e, f_s)$  e  $(s_p, f_k)$  permite converter  $f_s, f_k, s_e, s_p$  em  $f_1, f_2, \varepsilon_f, \eta$ . Para a conversão usa-se [Gafvert96]:

$$\begin{cases} f_1 = \frac{(m_1 m_2 + 2)p}{2(p-1)} f_k \\ f_2 = \frac{m_1 m_2 p + 2}{2(p-1)} f_k \\ \varepsilon_f = \frac{s_p}{3} \\ h = \frac{m_1 m_2 + 2}{m_1 m_2 p + 2} \end{cases}, \text{ com } \begin{cases} m_1 = \frac{f_s - f_k}{f_k} \\ m_2 = e^{3s_e/s_p} \end{cases} \quad (4.73)$$

onde  $p > 0 \cup p^{-1} - 1$  é dado pela solução da equação não linear  $\frac{m_1 m_2 + 2}{m_1 m_2} \ln(p) = (p - 1) \ln(m_2)$ .

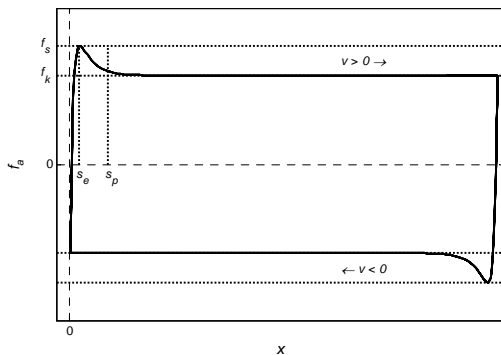


Figura 4.25 Força de atrito versus deslocamento, num movimento deslizante cíclico (com variação de velocidade) com histerese, para determinação dos parâmetros do modelo de 2ª ordem de Bliman e Sorine

De notar que o efeito de Stribeck não é o efeito de Stribeck clássico pois este só está presente a uma certa distância após o início do movimento, (Figura 4.25), não aparecendo quando o movimento abrandando, (Figura 4.26).

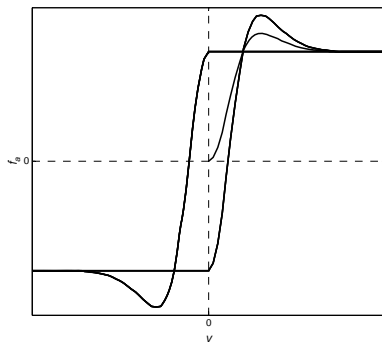


Figura 4.26 Força de atrito versus velocidade obtido por aplicação de uma velocidade de forma sinusoidal a uma massa deslizando com atrito modelado por um modelo de Bliman e Sorine de 2ª ordem

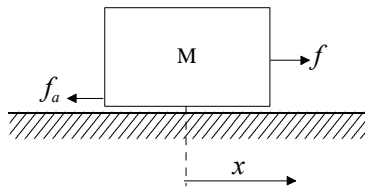


Figura 4.27 Sistema de massa livre com atrito

Nas figuras seguintes apresenta-se a simulação do sistema da Figura 4.27, onde a componente viscosa é inserida directamente na equação do movimento,  $f - f_a - k_v \times v = M \times \ddot{x}$ . Podem identificar-se os fenómenos de colagem, com características elásticas, e o movimento deslizando na fase inicial da simulação. Os parâmetros usados foram os seguintes:

$$f_s = 0.3019N, f_k = 0.2843N, s_e = 0.0037m, s_p = 0.0057m, k_v = 0Nsm^{-1}, M = 1Kg$$

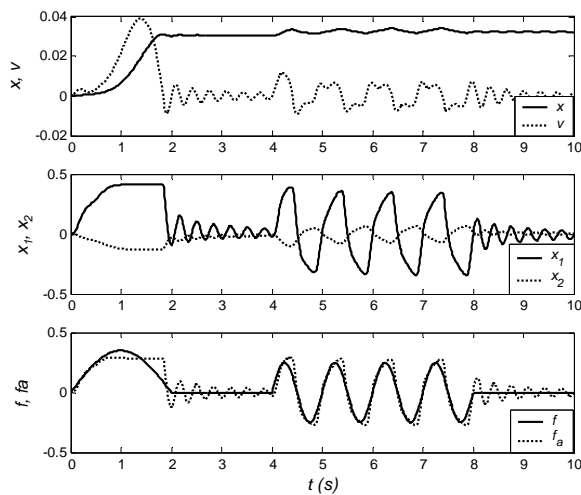


Figura 4.28 Aplicação de teste num sistema de massa livre sem componente de atrito viscoso. Podem ver-se os movimentos deslizando e pré-deslizante.

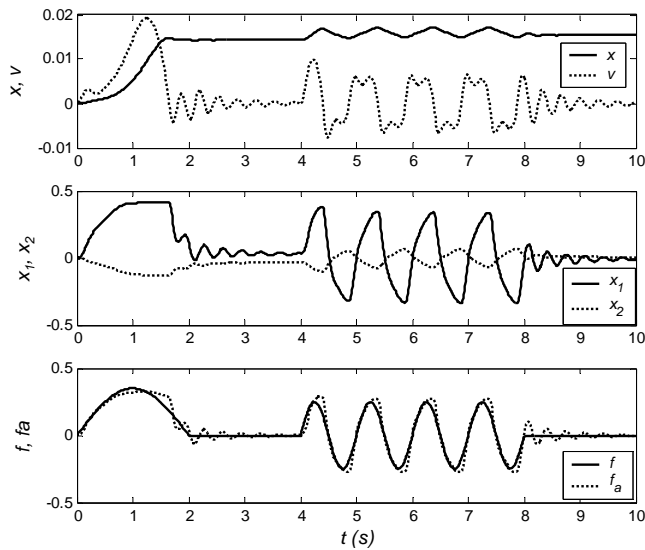


Figura 4.29 Aplicação de teste num sistema de massa livre com componente de atrito viscoso ( $k_v = 2Nsm^{-1}$ ).

## O modelo de LuGre

O modelo de LuGre [Canudas95] é um modelo dinâmico de atrito, que tem como entrada a velocidade relativa entre dois corpos em contacto e como saída a força de atrito. O modelo é inspirado na interpretação do atrito como resultante da força gerada pela “ligação” de um conjunto de “espetos” (rugosidades), salientes das duas superfícies em contacto, Figura 4.30. Estas ligações permitem a parametrização do modelo e ainda algumas interpretações físicas. O modelo de LuGre pode ser visto como uma extensão do modelo de Dahl. A deformação elástica, correspondente a pequenos deslocamentos, é descrita através do comportamento tipo mola dos “espetos” e resulta da aplicação de uma força tangencial. Se a força for suficientemente grande, alguns “espetos” deformam-se demasiado, tendo como consequência o deslizamento entre eles. Para grandes deslocamentos, a deformação plástica e a ruptura são descritas pelo deslocamento das ligações entre os “espetos”. A velocidade relativa entre as superfícies determina a quantidade de lubrificante que é forçado a entrar entre elas e, portanto, a distância de separação entre as superfícies.

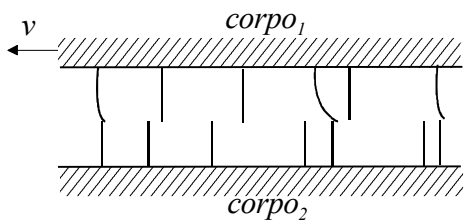


Figura 4.30 Atrito entre superfícies interpretado como resultante da “ligação” entre “espetos” pertencentes às duas superfícies. Para simplificação, os “espetos” do *corpo2* estão representadas de forma rígida [Olsson96]

O modelo de LuGre dá uma nova formulação ao modelo dos “espetos”, agregando o comportamento dos “espetos” num único “espeto”, como se ilustra na Figura 4.31, e adicionando algumas características extra. Neste caso, o “espeto” representa a deformação média de todos os “espetos” e, portanto, nunca pode descolar. Por exemplo, para um movimento com velocidade constante, a deformação do “espeto”,  $z$ , deve atingir um valor estacionário.

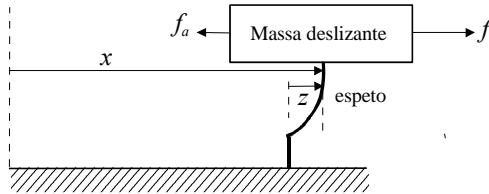


Figura 4.31 Um único “espeto” é usado para a representação da deformação média de todos os “espetos” em contacto

A deformação média dos “espetos” é representada pela variável de estado  $z$ , definida pela expressão:

$$\frac{dz}{dt} = v - \frac{|v|}{g\langle v \rangle} z \quad (4.74)$$

em que  $v$  é a velocidade relativa entre os corpos em contacto e  $g\langle v \rangle$  é uma função sempre positiva, dependente da velocidade.

A força de atrito é proporcional à deformação média e depende também da taxa de variação da deformação média. Além disso, o efeito do lubrificante também pode ser inserido através de uma função dependente da velocidade. Sendo assim, a força de atrito é dada por:

$$f_a = s_0 z + s_1 \langle v \rangle \frac{dz}{dt} + f\langle v \rangle \quad (4.75)$$

onde  $s_0 > 0$  é a rigidez média dos “espetos” e  $s_1 \langle v \rangle$  é o coeficiente de amortecimento, que pode ser dependente da velocidade. Para o caso mais típico de atrito viscoso pode considerar-se que a função  $f\langle v \rangle$  é linear com a velocidade:

$$f\langle v \rangle = k_v \times v \quad (4.76)$$

A força de amortecimento é muito importante para o comportamento do sistema na fase transitória entre a colagem e o deslizamento, pois atenua as oscilações na deformação do “espeto”. A parametrização linear do amortecimento é a mais simples e considera o coeficiente de amortecimento constante.

$$s_1 \langle v \rangle = s_1 \quad (4.77)$$

Outra parametrização sugerida em [Olsson96] considera que o amortecimento diminui com o aumento da velocidade:

$$s_1 \langle v \rangle = s_1 e^{-\langle v \rangle / v_d} \quad (4.78)$$

com  $v_d$  constante.

As mudanças nas características do amortecimento devem-se ao aumento de lubrificante que é forçado para a interface com o aumento da velocidade. Os valores de  $s_1$  dependem da carga inercial do sistema. A utilização de um amortecimento constante pode tornar o modelo de LuGre não dissipativo pois, segundo [Olsson96], o modelo é dissipativo se:

$$0 \leq s_1 \langle v \rangle \leq 4 \times s_0 \frac{g \langle v \rangle}{|v|} \quad (4.79)$$

A função  $g \langle v \rangle$  define em que medida a deformação média depende da velocidade relativa entre as superfícies em contacto e influencia o regime estacionário e o regime dinâmico. Em  $g \langle v \rangle$  deve modelar-se o efeito do lubrificante, não necessariamente de forma simétrica. Uma escolha para  $g \langle v \rangle$ , que fornece uma boa aproximação do efeito de Stribeck, é a seguinte:

$$g \langle v \rangle = \frac{1}{s_0} \left( f_{co} + f_s e^{-\langle v \rangle / v_s} \right) \quad (4.80)$$

onde  $f_{co}, f_s$  e  $v_s$  são, respectivamente, a força de atrito de Coulomb, a força de Stribeck e a velocidade crítica de Stribeck. De notar que  $g \langle v \rangle$  deve ter a forma da curva da força de atrito versus velocidade em regime estacionário, uma vez que a rigidez  $s_0$  é assumida como constante.

Para velocidades constantes, a força de atrito em regime estacionário,  $f_{ass}$ , é dada por (4.74) e (4.75) quando  $\frac{dz}{dt} = 0$ :

$$f_{ass} = s_0 \times g \langle v \rangle \times \text{sign} \langle v \rangle + f \langle v \rangle \quad (4.81)$$

## O modelo elasto-plástico

Um dos problemas exibido por alguns dos modelos apresentados anteriormente, nomeadamente os modelos de Dahl e de LuGre, é que estes apresentam deriva (deslocamentos sem limites) quando sujeitos a pequenas variações de força abaixo da força de *break-away*. Este tipo de comportamento pode constatar-se quando se aplica a força  $f$ , representada na Figura 4.32, ao sistema de massa livre com atrito modelado pelo modelo de LuGre. A partir dos dois segundos, a força aplicada é



oscilatória e inferior à força de *break-away*, havendo no entanto uma deriva na posição que mostra que o movimento também possui uma componente plástica. Utilizaram-se os seguintes parâmetros:

$$f_{co} = 1N, f_s = 0.1N, v_s = 0.1ms^{-1}, k_v = 0, s_0 = 100Nm^{-1}, s_1 = 2Nms^{-1}, M = 1Kg$$

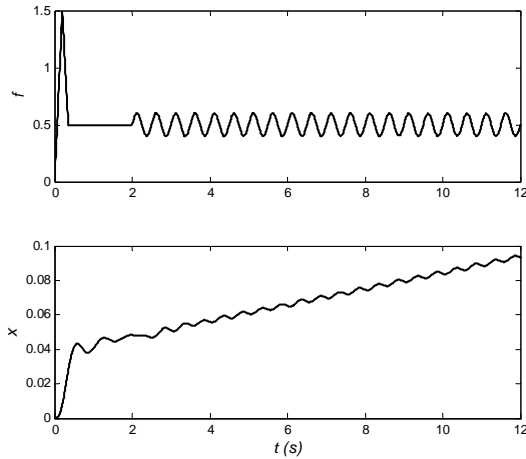


Figura 4.32 Deslocamento previsto utilizando o modelo de LuGre quando é aplicada uma força,  $f$ , ao sistema representado na Figura 4.27

Segundo Dupond [Dupond00], um modelo de atrito possui um “verdadeiro” estado de colagem quando existe uma força de *break-away* que, para qualquer força de atrito, se a expressão:

$$|f_a(t)| < f_{bw}, \quad \forall t < t_1 \quad (4.82)$$

for satisfeita então todos os movimentos correspondem a deformações elásticas e, portanto, são reversíveis. Quando a força deixa de ser aplicada, a posição regressa, após um período transitório, ao seu valor inicial.

Nos modelos dinâmicos de atrito com uma única variável de estado, o deslocamento,  $x$ , pode ser decomposto numa componente elástica,  $z$ , e numa componente plástica,  $d$ , esquematizadas na

Figura 4.33:

$$x = z + d \quad (4.83)$$

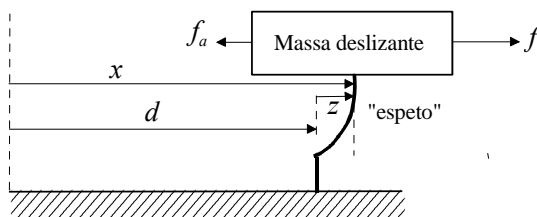


Figura 4.33 Massa deslizante com atrito, onde o deslocamento é representado por uma componente elástica (reversível) e uma componente plástica (irreversível)

Dupont, em [Dupont02], define três categorias para caracterizar o pré-deslizamento elasto-plástico:

$$\text{deslocamento elástico-} \begin{cases} \dot{x} = \dot{z} \\ d = 0 \end{cases} \quad (4.84)$$

$$\text{deslocamento misto (elástico e plástico)- } \dot{x} = \dot{z} + \dot{d} \quad (4.85)$$

$$\text{deslocamento plástico/deslizamento-} \begin{cases} \dot{x} = 0 \\ \dot{d} = \dot{z} \end{cases} \quad (4.86)$$

O pré-deslizamento elasto-plástico inicia-se num regime elástico (4.84), a que se segue um regime misto (4.85) e, depois, um regime plástico (4.86). Um pré-deslizamento plástico omite o regime elástico, enquanto que um modelo sem pré-deslizamento só possui o regime plástico.

Considere-se a curva de Stribeck que relaciona a força de atrito em regime estacionário,  $f_{ass}$ , com a velocidade:

$$f_{ass}(\dot{x}) = s_0 \times g(\dot{x}) \times \text{sign}(\dot{x}) \quad (4.87)$$

A deformação elástica em regime estacionário, correspondente a uma velocidade constante,  $\dot{x}$ , pode ser definida por:

$$z_{ss}(\dot{x}) = \begin{cases} \frac{f_{ass}(\dot{x})}{s_0}, & |\dot{x}| > 0 \\ \lim_{\dot{x} \rightarrow 0} \frac{f_{ass}(\dot{x})}{s_0}, & \dot{x} = 0 \end{cases} \quad (4.88)$$

enquanto que a força mínima e máxima em regime estacionário,  $f_{amax}$  e  $f_{amin}$ , são dadas por:

$$\begin{aligned} f_{a \max} &= \sup_{\dot{x}} |f_{ass}(\dot{x})| > 0 \\ f_{a \min} &= \inf_{\dot{x}} |f_{ass}(\dot{x})| > 0 \end{aligned} \quad (4.89)$$

A existência de um regime elasto-plástico pressupõe a existência de um valor de deformação,  $z_{ba}$ , tal que,  $|z| < z_{ba}$  implica que  $\dot{d} = 0$ , " $\dot{x} \in \mathbf{R}$ ".

O modelo plasto-elástico proposto em [Dupont00] e [Dupont02] propõe para a taxa de deformação do "espeto", do modelo de LuGre, a seguinte expressão:

$$\dot{z} = \frac{\dot{x}}{\sigma} \left[ 1 - a \left( \frac{z}{g(\dot{x})} \right) \frac{z}{g(\dot{x})} \text{sign}(\dot{x}) \frac{\ddot{x}}{\dot{x}} \right] \quad (4.90)$$

$a(z, \dot{z})$  é uma função de estrutura variável dada por:

$$a(z, \dot{z}) = \begin{cases} 0, & |z| < z_{ba} \\ 0 < a_m < 1, & z_{ba} \leq |z| < z_{ss} \\ 1, & |z| \geq z_{ss} \end{cases} \quad \text{sign}(\dot{z}) = \text{sign}(z) \quad (4.91)$$

onde  $0 < z_{ba} < z_{ss}$  e a função  $a_m$  controla o deslocamento misto e plástico, sendo definida por:

$$a_m(z, z_{ba}, z_{ss}) = \frac{1}{2} \sin\left(\frac{z - \frac{z_{ba} + z_{ss}}{2}}{z_{ss} - z_{ba}}\right) + \frac{1}{2} \quad (4.92)$$

A dependência de  $z_{ss}$  de  $\dot{z}$  deve-se, segundo Dupond, à utilização de, apenas, um estado no modelo de atrito para representação do pré-deslizamento e do efeito de Stribeck. É definido um deslocamento de *break-away*,  $z_{ba} > 0$ , tal que o modelo se comporta de forma elástica enquanto  $|z| < z_{ba}$ . O valor da deformação,  $z$ , também é limitado por  $|z| < z_{max} = f_{a,max}/s_0$ . De notar também que (4.74) pode ser obtido através de (4.90) para  $a(z, \dot{z}) = 1$ .

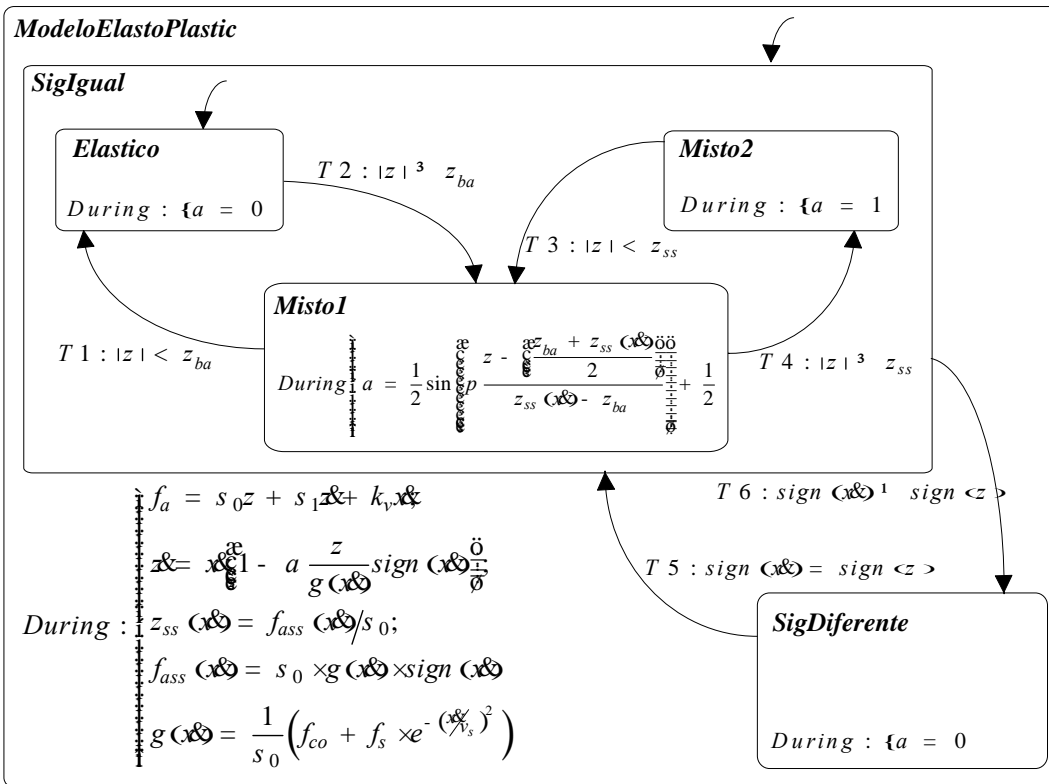


Figura 4.34 Statechart de implementação do modelo de atrito elasto-plástico

Foi repetida a experiência ilustrada na Figura 4.27, com o atrito a ser modelado pelo statechart da Figura 4.34. Os resultados, apresentados na Figura 4.35, mostram a existência de um estado elástico não havendo deriva, contrariamente ao verificado na simulação com atrito de LuGre apresentada na Figura 4.32. A simulação utilizou os seguintes parâmetros:

$$f_{co} = 1N, f_s = 0.1N, v_s = 0.1ms^{-1}, k_v = 0, s_0 = 100Nm^{-1}, s_1 = 2Nms^{-1}, z_{ba} = 7e^{-3}m, M = 1Kg$$

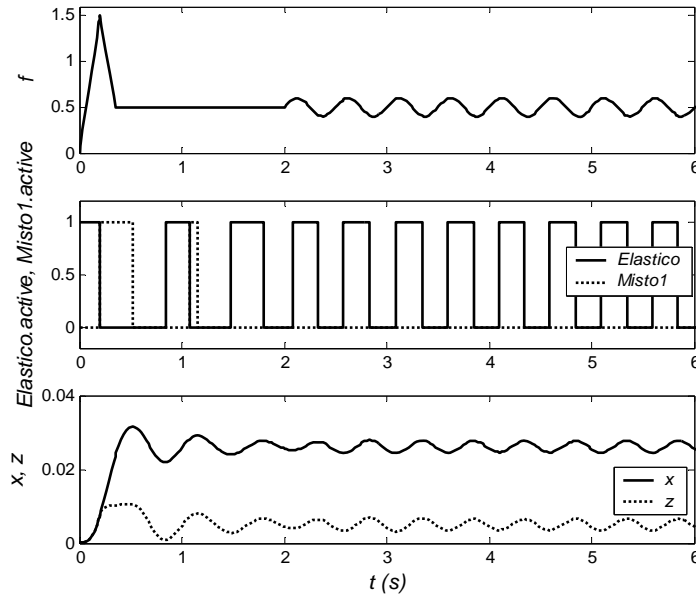


Figura 4.35 Simulação de uma massa deslizando com atrito elasto-plástico

Nesta secção, a problemática da modelação do atrito foi abordada através da apresentação de diferentes modelos. Destacaram-se as duas principais “categorias” de modelos: modelos estáticos e modelos dinâmicos. Os modelos de atrito híbridos foram implementados usando statecharts, para uma melhor compreensão do comportamento em simulação; por exemplo, o modelo de Armstrong evidencia as potencialidades do formalismo dos Statecharts para descrição de comportamentos híbridos.

Os modelos de atrito estáticos modelam as forças de atrito como sendo uma função da velocidade. Estes modelos contemplam o atrito de Coulomb, o atrito estático (força de *break-away*), o efeito de Stribeck e a componente viscosa do atrito (proporcional à velocidade). Um dos problemas, para alguns desses modelos, reside na indefinição da força de atrito quando a velocidade é nula. Nesse contexto, o modelo de Karnopp considera a velocidade nula quando esta está numa região próxima de zero e, nessa situação, a força de atrito é calculada em função das outras forças presentes no sistema. Na “fronteira”, entre os modelos de atrito estáticos e os dinâmicos, encontra-se o modelo de Armstrong; este contempla algumas propriedades que são dependentes do tempo como, por

exemplo, a influência do tempo de permanência, num estado de colagem, no nível do atrito estático e um atraso temporal, entre uma variação de velocidade e a correspondente variação na força de atrito. Um dos problemas, na implementação deste modelo, é o elevado número de parâmetros que é necessário identificar.

Na outra categoria, os modelos de atrito dinâmicos têm particular interesse na modelação de servomecanismos de precisão e no desenvolvimento de compensadores de atrito. A diferença entre os modelos estáticos e os modelos dinâmicos reside na utilização de variáveis de estado, normalmente uma ou duas. O modelo de Dahl é modelo dinâmico mais simples apresentado; contudo, este não modela a colagem nem uma das características estáticas importantes como é caso do efeito de Stribeck. Os modelos de Bliman-Sorine e de LuGre podem considerar-se extensões do modelo de Dahl. Por exemplo, o modelo de 2ª ordem de Bilman-Sorine usa dois modelos de Dahl, com dinâmicas diferentes, para descrever o estado de colagem. Estes dois modelos apresentam, contudo, algumas diferenças (apresentadas com detalhe em [Gafvert96]): por exemplo, o modelo de Bliman-Sorine apresenta um comportamento oscilatório, na força de atrito, a baixas velocidades; o modelo de LuGre necessita que a condição  $s_1 < 4g \langle v \rangle / |v|$  se verifique para que o modelo seja dissipativo, sendo então desejável que  $s_1$  seja dependente da velocidade. Dos modelos dinâmicos apresentados, o modelo elasto-plástico apresenta-se como o mais completo. Este define um comportamento semelhante ao modelo de LuGre. A principal diferença reside na descrição de um comportamento verdadeiramente elástico, no estado de colagem, à custa da introdução de uma nova variável de estado. O modelo de LuGre apresenta uma boa relação desempenho/complexidade, o que, em conjunto com o baixo “custo” na identificação dos parâmetros, pode considerar-se uma boa solução para a modelação do fenómeno do atrito em sistemas hidráulicos.

## 5 MODELOS PARA VÁLVULAS PROPORCIONAIS

O hardware hidráulico sofreu uma grande evolução nos últimos anos. A introdução da electrónica e dos microprocessadores permitiu a melhoria do desempenho dos componentes hidráulicos, em particular das válvulas proporcionais. O avanço da tecnologia das válvulas electro-hidráulicas proporcionais permite, actualmente, a sua concorrência com as servoválvulas em algumas aplicações. Hoje em dia, as válvulas proporcionais de elevado desempenho podem suportar algumas aplicações em malha fechada, anteriormente só possíveis com a utilização de servoválvulas, com custos e necessidades de manutenção mais reduzido.

A variável de entrada, típica de um servomecanismo hidráulico, é uma força aplicada à gaveta de uma válvula. Esta força faz variar a posição da gaveta e, por conseguinte, as áreas dos orifícios de passagem que controlam os escoamentos e as pressões do sistema. Quando a válvula é do tipo proporcional a força de entrada é normalmente produzida por um solenóide proporcional. Este tipo de solenóide produz, dentro de uma determinada região de operação, uma força proporcional à intensidade de corrente eléctrica que o percorre. O movimento da gaveta está dependente das perturbações resultantes das forças de atrito entre a gaveta e a camisa da válvula, das forças geradas pelas diferentes pressões devidas à geometria da gaveta, e de forças inerciais induzidas pelos escoamentos nos orifícios. As perturbações no movimento da gaveta são altamente indesejáveis e limitam a utilização das válvulas proporcionais, ditas comuns, em operações de controlo em malha fechada. Nas válvulas proporcionais de elevado desempenho, também chamadas válvulas servo-solenóide, são usadas estratégias avançadas de controlo em malha fechada da posição da gaveta, através de hardware dedicado, que permitem melhorar as características dinâmicas ao longo da região de operação da válvula. Deste modo, podem ser minimizadas as dinâmicas altamente não lineares inerentes ao mecanismo de movimento da gaveta. Assim, a resposta na frequência é aumentada até, tipicamente, 200 Hz, para variações do sinal de entrada de  $\pm 5\%$ , numa válvula NG6 (caudal nominal até 40 *l/min*, para uma pressão nominal de 35 *bar* por orifício). A Figura 5.1 mostra uma válvula com a carta de controlo integrada.

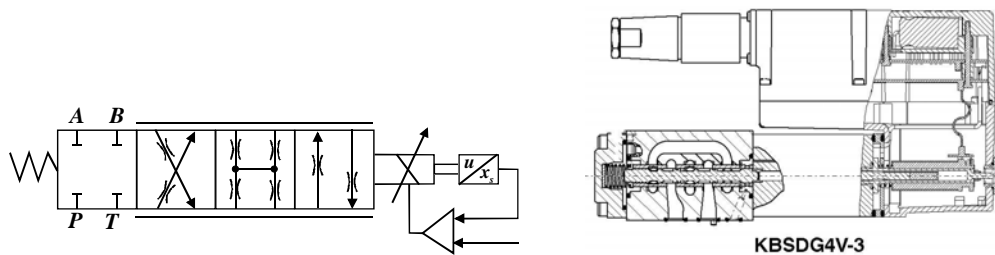


Figura 5.1 Simbologia e corte de uma válvula proporcional modelo KBSDG4V-3 da Vickers (Cortesia da Eaton Corporation)

Vários modelos para válvulas proporcionais podem ser encontrados na literatura [Merrit67], [McCloy73], [Lebrun87], [Watton89], [Handroos91], [Vaughan96], [Virtalo97], [Ellman98], [Elmer99], [Feky99], [Quintas99], [Borghi00] ou [Koskinen00]. Alguns autores descrevem modelos teóricos complexos, usando técnicas da mecânica de fluidos para a modelação das forças devidas ao escoamento, outros modelam as características das molas, forças de atrito no órgão móvel e a própria geometria da válvula no regime estacionário e em regime dinâmico. Contudo, e apesar da grande evolução do hardware para cálculo científico, a simulação em tempo real destes modelos teóricos complexos é, ainda hoje, praticamente impossível. Outra grande dificuldade é a identificação da geometria da válvula e dos seus parâmetros físicos, necessários à parametrização dos modelos, informação essa raramente fornecida pelos fabricantes.

Outros modelos propostos para válvulas proporcionais são os chamados modelos semi-empíricos. Estes requerem normalmente menores recursos computacionais e os seus parâmetros são de ajuste mais fácil, através de cálculos numéricos simples, utilizando características medidas ou fornecidas pelo fabricante. Estes modelos capturam as características estáticas e dinâmicas mais influentes, utilizando funções matemáticas bem conhecidas, e são normalmente independentes das pequenas variações geométricas que ocorrem no processo de produção.

Os volumes das câmaras das válvulas são bastantes pequenos, quando comparados com os volumes das condutas e das câmaras dos actuadores. Deste modo, os escoamentos devidos à compressibilidade do fluido hidráulico são geralmente inseridos nos modelos das condutas ou dos actuadores [Merrit67]. Assim, se os caudais de transição forem desprezáveis, em cada instante, durante o movimento da gaveta, o caudal de saída é o mesmo que o obtido no regime estacionário para a mesma posição da gaveta e as mesmas condições de pressão [Brun99]. Os caudais através da válvula podem então ser modelados por equações algébricas.

Sendo assim, a parte dinâmica do modelo pode ser separada da parte estática e os modelos das válvulas podem ser divididos em dois blocos principais, com uma ligação série entre eles (como ilustra a Figura 5.2). O primeiro bloco tem como entrada o sinal eléctrico relativo de referência

para a posição da gaveta ( $\bar{u} \in [-1, 1]$ ), como saída a posição real relativa da gaveta ( $\bar{x}_s \in [-1, 1]$ ), e modela a dinâmica não linear do movimento da gaveta. O segundo bloco descreve o comportamento estático da válvula e relaciona os caudais através da válvula ( $q_i$ ) com a posição relativa da gaveta,  $\bar{x}_s$ , e as pressões nos terminais da válvula,  $P_i$ .

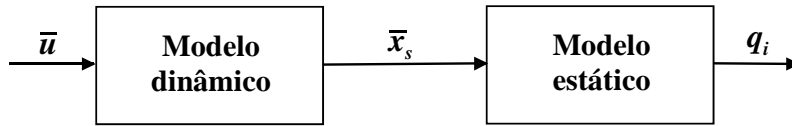


Figura 5.2 Desacoplamento da parte dinâmica e estático do modelo de válvula

Nas próximas secções apresentam-se alguns dos modelos, propostos neste trabalho, para a parte estática e dinâmica de válvulas do tipo servo-solenóide.

## 5.1 Modelos estáticos

Os modelos estáticos propostos pretendem reproduzir as características estáticas da válvula (valores experimentais ou fornecidos pelo fabricante), tais como o ganho de pressão relativo, o caudal de fugas e o ganho de caudal, especialmente perto da posição central da válvula, pois os sistemas geralmente operam nessa região. O ganho de caudal nesta posição afecta a constante de ganho em malha aberta, enquanto as características do ganho de pressão e o caudal de fugas estão ligadas à rigidez do servomecanismo. Pode-se considerar, sem perda de generalidade, que este tipo de válvulas tem quatro secções de controlo moduladas pela posição da gaveta,  $\bar{x}_s$ , como se pode ver na Figura 5.3 e na Figura 5.4.

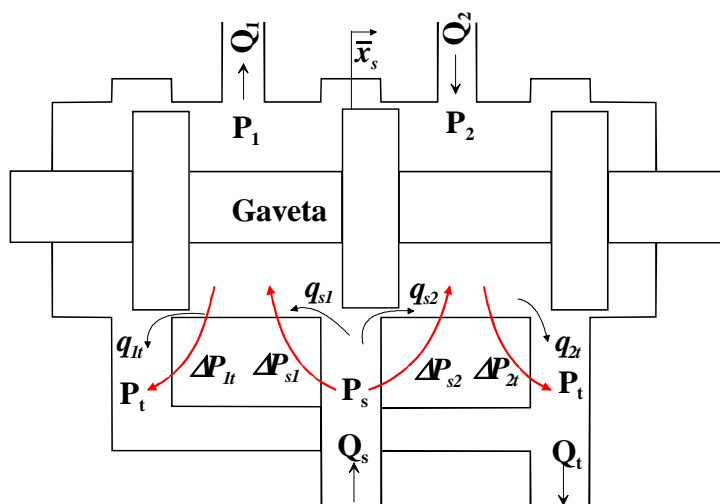


Figura 5.3 Diagrama de uma válvula de 4 orifícios



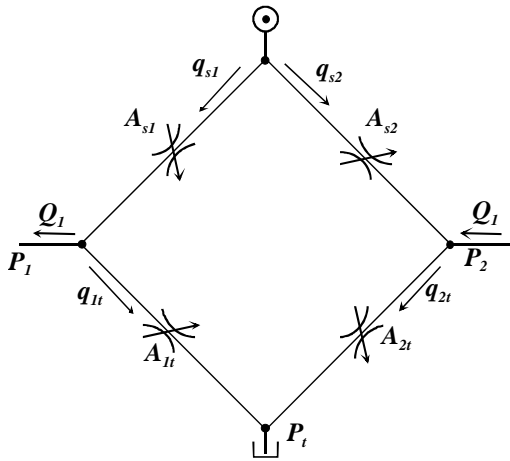


Figura 5.4 Representação em Ponte de Weathstone de uma válvula de 4 orifícios

O caudal  $q$ , correspondente ao escoamento turbulento, através de um orifício é definido da seguinte forma [Merrit67], [McCloy73], [Watton89]:

$$q = \text{sign}(DP) \times C_d A_0 \times \sqrt{\frac{2}{r} |DP|} \quad (5.1)$$

onde  $C_d$  é o coeficiente de descarga,  $A_0$  é a área do orifício,  $DP$  é a diferença de pressão entre a entrada e a saída do orifício,  $\text{sign}(DP)$  é o sinal de  $DP$  e  $r$  é a massa específica do fluido.

Seja  $A_{ij}(\bar{x}_s)$  uma pseudo-seção que relaciona o escoamento turbulento,  $q_{ij}$ , desde o terminal  $i$  para o terminal  $j$ , com a diferença de pressão entre os terminais. A equação de orifício (5.1) pode então ser re-escrita:

$$q_{ij} = \text{sign}(DP_{ij}) \times A_{ij}(\bar{x}_s) \times \sqrt{|DP_{ij}|} \quad (5.2)$$

onde  $DP_{ij} = P_i - P_j$  é a diferença de pressão entre os dois terminais.

O caudal correspondente ao escoamento laminar é dado por:

$$q_{ij\text{lam}} = g_{lk}(\bar{x}_s) \times DP_{ij} \quad (5.3)$$

onde  $g_{lk}(\bar{x}_s)$  é a condutância de fugas em função da posição relativa da gaveta,  $\bar{x}_s$ .

As válvulas proporcionais de elevado desempenho, como as servo-válvulas, são normalmente fabricadas com orifícios de controlo simétricos e emparelhados. Esta opção, segundo Merrit [Merrit67], leva ao máximo ganho de malha e rigidez na carga, quando comandam actuadores simétricos. A válvula tem orifícios emparelhados se  $A_{s1}(\bar{x}_s) = A_{t2}(\bar{x}_s) = A_p(\bar{x}_s)$  e

$A_{s2}(\bar{x}_s) = A_{1t}(\bar{x}_s) = A_n(\bar{x}_s)$ ; a simetria significa que  $A_p(\bar{x}_s) = A_n(-\bar{x}_s)$ . Nestas condições de simetria e emparelhamento, as condutâncias de fugas, a considerarem-se, relacionam-se de igual modo, isto é:  $g_{ks1}(\bar{x}_s) = g_{lk2t}(\bar{x}_s) = g_{lkp}(\bar{x}_s)$  e  $g_{ks2}(\bar{x}_s) = g_{lk1t}(\bar{x}_s) = g_{lkn}(\bar{x}_s)$  e, sendo a válvula simétrica,  $g_{lkp}(\bar{x}_s) = g_{lkn}(-\bar{x}_s)$ .

De acordo com a nomenclatura apresentada nas Figura 5.3 e Figura 5.4, as equações genéricas (contendo uma componente turbulenta e uma componente laminar) para os caudais através da válvula podem ser escritas como:

$$\begin{aligned} \dot{q}_{s1} &= \text{sign}(DP_{s1}) \times A_{s1}(\bar{x}_s) \times \sqrt{|DP_{s1}|} + g_{ks1}(\bar{x}_s) \times DP_{s1} \\ \dot{q}_{s2} &= \text{sign}(DP_{s2}) \times A_{s2}(\bar{x}_s) \times \sqrt{|DP_{s2}|} + g_{ks2}(\bar{x}_s) \times DP_{s2} \\ \dot{q}_{1t} &= \text{sign}(DP_{1t}) \times A_{1t}(\bar{x}_s) \times \sqrt{|DP_{1t}|} + g_{lk1t}(\bar{x}_s) \times DP_{1t} \\ \dot{q}_{2t} &= \text{sign}(DP_{2t}) \times A_{2t}(\bar{x}_s) \times \sqrt{|DP_{2t}|} + g_{lk2t}(\bar{x}_s) \times DP_{2t} \end{aligned} \quad (5.4)$$

Para uma válvula simétrica e centrada, com os orifícios completamente emparelhados (caso mais usual), o número de pseudo-seções pode ser reduzido. Desta forma, assumindo emparelhamento e simetria, as equações para os caudais através da válvula, (5.4), podem ser simplificadas:

$$\begin{aligned} \dot{q}_{s1} &= \text{sign}(DP_{s1}) \times A_p(\bar{x}_s) \times \sqrt{|DP_{s1}|} + g_{lkp}(\bar{x}_s) \times DP_{s1} \\ \dot{q}_{s2} &= \text{sign}(DP_{s2}) \times A_n(\bar{x}_s) \times \sqrt{|DP_{s2}|} + g_{lkn}(\bar{x}_s) \times DP_{s2} \\ \dot{q}_{1t} &= \text{sign}(DP_{1t}) \times A_n(\bar{x}_s) \times \sqrt{|DP_{1t}|} + g_{lkn}(\bar{x}_s) \times DP_{1t} \\ \dot{q}_{2t} &= \text{sign}(DP_{2t}) \times A_p(\bar{x}_s) \times \sqrt{|DP_{2t}|} + g_{lkp}(\bar{x}_s) \times DP_{2t} \end{aligned} \quad (5.5)$$

Tomando como referência o diagrama da Figura 5.4, o caudal para o tanque,  $Q_t$ , o caudal da fonte,  $Q_s$ , e os caudais de saída,  $Q_1$  e  $Q_2$ , podem ser formulados da seguinte maneira:

$$\begin{aligned} \dot{Q}_s &= q_{s1} + q_{s2} \\ \dot{Q}_t &= q_{1t} + q_{2t} \\ \dot{Q}_1 &= q_{s1} - q_{1t} \\ \dot{Q}_2 &= q_{2t} - q_{s2} \end{aligned} \quad (5.6)$$

Quando a válvula está ligada a um actuador simétrico, os caudais de saída são iguais, *i.e.*  $Q_1 = Q_2$ . Se a pressão no tanque for, sem perda de generalidade, assumida como igual a zero (ou então  $P_t$  for considerada a pressão de referência para todas as outras pressões), as pressões nos terminais estão relacionadas por [Almeida93]:

$$P_s = P_1 + P_2 \quad (5.7)$$

### 5.1.1 Características estáticas da válvula

#### Ganho de caudal

O ganho de caudal é definido como a derivada do caudal na carga em relação a  $\bar{x}_s$ , com diferença de pressão nula na carga, na posição intermédia da gaveta ( $\bar{x}_s = 0$ ):

$$\bar{K}_{q0} = \left. \frac{dQ_L}{d\bar{x}_s} \right|_{\bar{x}_s=0} \quad (5.8)$$

O caudal na carga,  $Q_L$ , quando os terminais 1 e 2 forem ligados por uma resistência nula, é dado por:

$$Q_L = Q_1 = q_{s1} - q_{1t} = Q_2 = q_{2t} - q_{s2} \quad (5.9)$$

sendo as pressões  $P_1$  e  $P_2$  iguais a  $\frac{P_s}{2}$ .

#### Ganho de pressão relativo

A pressão na carga é definida como a diferença de pressão entre os terminais 1 e 2 ( $P_L = P_1 - P_2$ ).

A pressão relativa na carga é definida como  $\bar{P}_L = P_L/P_s$ . O ganho de pressão relativo é definido como a derivada da pressão relativa na carga em relação a  $\bar{x}_s$ , na posição intermédia da gaveta ( $\bar{x}_s = 0$ ), quando o caudal na carga é nulo ( $Q_L = 0$ ):

$$\bar{K}_{p0} = \left. \frac{d\bar{P}_L}{d\bar{x}_s} \right|_{\bar{x}_s=0} \quad (5.10)$$

Usando (5.7), a pressão nos terminais pode ser expressa em função de  $\bar{P}_L$ :

$$\begin{cases} P_1 = P_s - P_2 = \frac{P_s}{2}(1 + \bar{P}_L) \\ P_2 = P_s - P_1 = \frac{P_s}{2}(1 - \bar{P}_L) \end{cases} \quad (5.11)$$

Da expressão  $Q_L = Q_L(\bar{x}_s, \bar{P}_L)$ , e no caso da configuração ilustrada na Figura 5.6, pode obter-se uma expressão para  $\bar{P}_L$ , já que o caudal na carga é nulo e os caudais internos vão ser iguais dois a dois ( $q_{1t} = q_{s1}$  e  $q_{2t} = q_{s2}$ ).

#### Caudal de fugas

Considerando a montagem ilustrada na Figura 5.7, o caudal de fugas ( $q_{lk}$ ) pode ser dado por:

$$q_{lk} = q_{1t} + q_{2t} = q_{s1} + q_{s2} = q_{1t} + q_{s2} = q_{s1} + q_{2t} \quad (5.12)$$

O caudal de fugas na posição central, *i.e.*  $\bar{x}_s = 0$ ,  $\bar{P}_L = 0$ , vem dado por:

$$q_{lk0} = q_{lk}(\bar{x}_s, \bar{P}_L) \Big|_{\substack{\bar{x}_s=0 \\ \bar{P}_L=0}} \quad (5.13)$$

### Medição das características estáticas da válvula

Para a medição das características das válvulas seguiu-se a metodologia proposta em [Quintas99] (o esquema de ligações para medição das características de caudal na carga é ilustrado na Figura 5.5). Idealmente, as características de caudal na carga deveriam ser medidas para uma carga nula, de modo a que as pressões nos terminais 1 e 2 fossem iguais a  $P_n$  (pressão nominal, por orifício) e fixando  $P_s=2 P_n$ , de modo a que as diferenças de pressão em todos os orifícios da válvula fossem constantes e iguais a  $P_n$ . Como tal não é normalmente possível, pois o transdutor de caudal não tem resistência hidráulica nula, ajusta-se a pressão na fonte,  $P_s$ , com um factor  $\Delta P_{12}$  de modo a manter constante a soma das quedas de pressão nos orifícios da válvula. A medição das características de pressão foram realizadas nas condições da montagem ilustrada na Figura 5.6. Para a realização das medições tamponaram-se os terminais 1 e 2 de modo a anular o caudal na carga, medindo-se as características de pressão  $P_1$  e  $P_2$ , para  $P_s=2 P_n$  e para todo o curso da gaveta. Para a medição do caudal de fugas utilizou-se um transdutor de caudal para a medição do caudal para o tanque  $Q_t$ , Figura 5.7, que nesta montagem é o caudal de fugas. Para a medição dos caudais nos diferentes orifícios da válvula podem usar-se as montagens propostas por Quintas, em [Quintas99]. Com estas medições é possível obter os parâmetros estáticos mais importantes para a caracterização da válvula: ganho de caudal, caudal nominal, ganho de pressão e caudal de fugas.

As medições foram realizadas através de um transdutor de caudal de engrenagens, da Webster Instruments®, que gera um sinal eléctrico cuja frequência é proporcional ao caudal medido. O sinal gerado foi, através de um dispositivo electrónico desenvolvido para o efeito, convertido num sinal em tensão entre 0 e 10 V, de modo a ser adquirido pelo hardware de aquisição de dados.

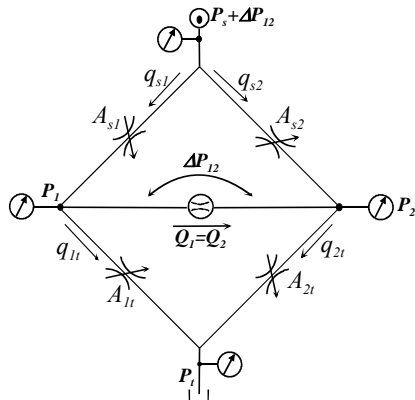


Figura 5.5 Montagem para medição das características de caudal na carga

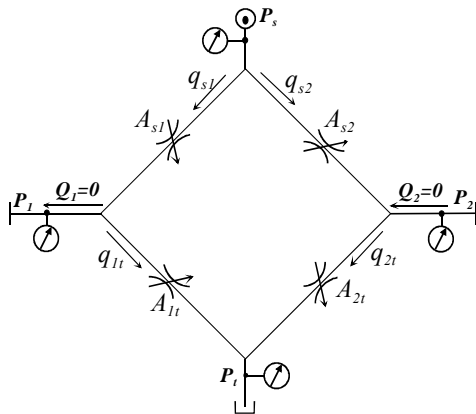


Figura 5.6 Montagem para a medição das características da pressão na carga

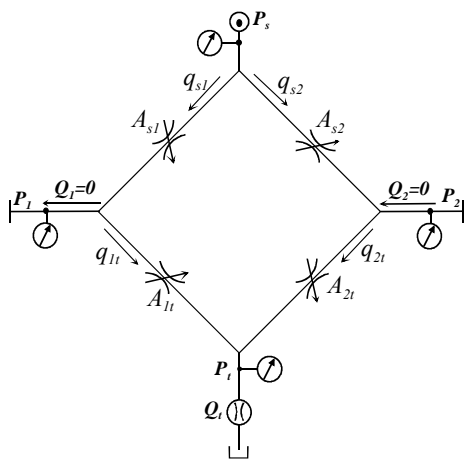


Figura 5.7 Montagem para a medição do caudal de fugas

Os fabricantes de válvulas fornecem, geralmente, a seguinte informação: característica de pressão na carga (para caudal na carga nulo); uma estimativa do caudal de fugas máximo; o caudal nominal à diferença de pressão nominal; uma estimativa do ganho de caudal perto da origem (com, tipicamente, uma incerteza compreendida entre 50% e 200% do ganho de caudal nominal); os diagramas de Bode para diferentes amplitudes no movimento da gaveta. Os parâmetros para os modelos da válvula podem ser ajustados através dos manuais do fabricante ou através de medições experimentais. Uma válvula proporcional da Vickers, modelo KBSDG4V-3 [Vickers99], é usada nas secções seguintes como um exemplo para o cálculo dos parâmetros. As curvas da pressão relativa na carga, do caudal na carga e do caudal de fugas podem ser usadas para extrair os parâmetros do modelo (caudal nominal, ganho de caudal, ganho de pressão e caudal de fugas na posição intermédia). As simulações apresentadas a seguir, salvo indicação em contrário, usam as seguintes características medidas:

$$Q_n = 25.5 \text{ l/min e } \bar{K}_{q0} = 28 \text{ l/min a } P_n = 35 \text{ bar, } \bar{K}_{p0} = 36.5 \text{ e } q_{i0} = 1.36 \text{ l/min a } P_s = 70 \text{ bar.}$$

Seguiram-se duas filosofias no desenvolvimento dos diferentes modelos para válvulas proporcionais apresentados neste trabalho: orifícios com escoamento misto (turbulento e laminar) e orifícios somente com escoamento turbulento. Para todos os modelos, propostos nas secções seguintes, apresentam-se os métodos de parametrização com base em algumas características estáticas: ganho de caudal, ganho de pressão, caudal nominal e caudal de fugas.

### Funções para as pseudo-secções

De seguida averigua-se a possibilidade de ajuste simultâneo das três características estáticas da válvula (ganho de pressão, ganho de caudal e caudal de fugas), na posição central da gaveta.

Na posição central, a equação  $\bar{x}_s = \bar{x}_s(Q_L, \bar{P}_L(Q_L)) = 0$  define uma função implícita entre  $\bar{P}_L$  e  $Q_L$  e a equação (5.14) pode ser usada para a válvula de quatro orifícios em consideração.

$$\frac{\partial Q_L}{\partial \bar{x}_s} = - \frac{\partial Q_L}{\partial \bar{P}_L} \frac{\partial \bar{P}_L}{\partial \bar{x}_s} \quad (5.14)$$

O coeficiente caudal-pressão relativa,  $-\frac{\partial Q_L}{\partial \bar{P}_L}$ , pode ser encontrado por diferenciação da equação em  $Q_L$ . A partir de (5.5), (5.6), (5.9) e (5.11) pode ser escrita a seguinte equação para  $Q_L$ :

$$Q_L = A_p(\bar{x}_s) \sqrt{\frac{P_s}{2}(1 - \bar{P}_L)} - A_n(\bar{x}_s) \sqrt{\frac{P_s}{2}(1 + \bar{P}_L)} + g_{lkp}(\bar{x}_s) \frac{P_s}{2}(1 - \bar{P}_L) - g_{lkn}(\bar{x}_s) \frac{P_s}{2}(1 + \bar{P}_L) \quad (5.15)$$

Da mesma forma o caudal de fugas, (5.12), pode ser expresso por:

$$q_{lk} = q_{lkT} + q_{lkL} = \begin{cases} 2A_p(\bar{x}_s) \sqrt{\frac{P_s}{2}(1 - \bar{P}_L)} + g_{lkp}(\bar{x}_s) P_s (1 - \bar{P}_L) \\ \text{ou} \\ 2A_n(\bar{x}_s) \sqrt{\frac{P_s}{2}(1 + \bar{P}_L)} + g_{lkn}(\bar{x}_s) P_s (1 + \bar{P}_L) \end{cases} \quad (5.16)$$

onde  $q_{lkT}$  e  $q_{lkL}$  são os caudais de fugas relativos aos escoamentos turbulentos e laminares, respectivamente.

Derivando  $Q_L$  em relação a  $\bar{P}_L$  tem-se:

$$\frac{\partial Q_L}{\partial \bar{P}_L} = - \frac{1}{4} \frac{A_p(\bar{x}_s)}{\sqrt{\frac{P_s}{2}(1 - \bar{P}_L)}} P_s - \frac{1}{4} \frac{A_n(\bar{x}_s)}{\sqrt{\frac{P_s}{2}(1 + \bar{P}_L)}} P_s - \frac{1}{2} g_{lkp}(\bar{x}_s) P_s - \frac{1}{2} g_{lkn}(\bar{x}_s) P_s \quad (5.17)$$

Quando  $\bar{x}_s = 0$ , nas condições de medida do ganho de pressão ou do ganho de caudal,  $\bar{P}_L = 0$ . Então, usando (5.14), (5.16) e (5.17), a relação entre o ganho de caudal, ganho de pressão relativo e caudal de fugas, na posição intermédia da gaveta, resulta em:

$$\left. \frac{Q_L}{\bar{x}_s} \right|_{\bar{x}_s=0} = \left( \frac{q_{lkT0}}{2} + q_{lkL0} \right) \left. \frac{\bar{P}_L}{\bar{x}_s} \right|_{\bar{x}_s=0} \quad (5.18)$$

onde o caudal de fugas na posição intermédia é dado por

$$q_{lk0} = q_{lkT0} + q_{lkL0} \quad (5.19)$$

Isto significa que o modelo, a existirem funções para as pseudo-secções e para as condutâncias de fugas que respeitem, por exemplo, a relação:

$$q_{lkT0} = 2 \frac{K_{q0}}{K_{p0}} q_{lk0} - \frac{K_{q0}}{K_{p0}} \frac{\bar{P}_L}{\bar{P}_0} \quad (5.20)$$

pode ajustar simultaneamente as três características estáticas da válvula.

As características estáticas usadas para o cálculo dos parâmetros são: o caudal nominal ( $Q_n$ ) para uma diferença de pressão igual à pressão nominal  $P_n$ ; ganho de pressão relativa; caudal de fugas ou o ganho de caudal na posição intermédia.

## 5.1.2 Modelos com escoamento misto

Nos modelos apresentados nesta secção considera-se que cada orifício tem uma componente de escoamento laminar, caracterizado por uma condutância de fugas dependente da posição relativa da gaveta, e uma componente de escoamento turbulento em que a secção do orifício varia linearmente com a posição relativa da gaveta. Estas simplificações constam dos manuais de fabricantes deste tipo de válvulas para a caracterização do caudal.

### Modelo estático 1

Neste modelo a condutância de fugas ( $g_{lk}$ ) versus posição relativa da gaveta ( $\bar{x}_s \in [-1, 1]$ ) é modelada por uma função exponencial:

$$g_{lk} = g_{lkp} = g_{lk} = \begin{cases} g_{lk} e^{l(\bar{x}_s + \bar{r})} & ; \bar{x}_s < -\bar{r} \\ g_{lk} & ; -\bar{r} \leq \bar{x}_s \leq \bar{r} \\ g_{lk} e^{-l(\bar{x}_s - \bar{r})} & ; \bar{x}_s > \bar{r} \end{cases} \quad (5.21)$$

$g_{lkm}$  é a condutância de fugas máxima,  $l$  define o decaimento da curva exponencial e  $r$  define o recobrimento negativo da válvula. Considera-se que a condutância de fugas da válvula é máxima quando a gaveta da válvula se encontra na região de recobrimento negativo,  $\bar{r}$ . Fora dessa região a condutância é modelada por uma função baseada na função exponencial onde a constante  $l$  define o decaimento, Figura 5.8. A ideia subjacente a este modelo tem a ver com a existência de três parâmetros, para averiguar se é possível ajustar os três parâmetros estáticos relevantes neste tipo de válvulas: ganho de pressão relativa, ganho de caudal e caudal de fugas máximo na origem.

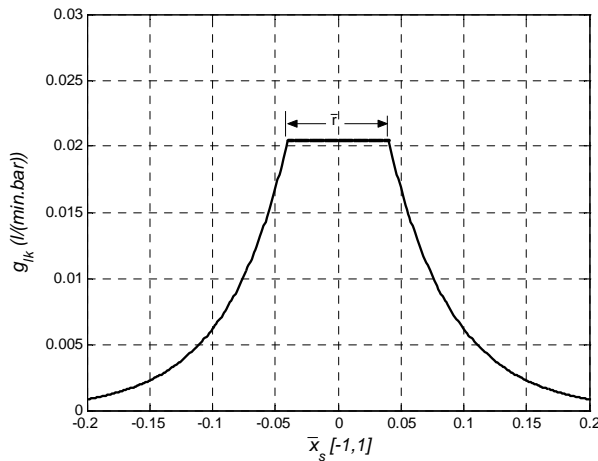


Figura 5.8 Característica da condutância de fugas em função da posição relativa da gaveta

As secções, usadas em regime turbulento, consideram-se proporcionais à abertura do orifício:

$$A_{s1} = A_{2t} = A_p \begin{cases} (\bar{x}_s + \bar{r}) \times cte & ; \bar{x}_s \geq -\bar{r} \\ 0 & ; \bar{x}_s < -\bar{r} \end{cases}$$

$$A_{s2} = A_{1t} = A_n = \begin{cases} (\bar{r} - \bar{x}_s) \times cte & ; \bar{x}_s < \bar{r} \\ 0 & ; \bar{x}_s \geq \bar{r} \end{cases} \quad (5.22)$$

$cte$  é uma constante que depende do caudal nominal ( $Q_n$ ) e da pressão nominal ( $P_n$ ) da válvula,:

$$cte = \frac{Q_n}{\sqrt{P_n}} \quad (5.23)$$

### Ganho de caudal:

Analisando para  $\bar{x}_s > 0$ , e considerando  $P_s = 2 P_n = 2 P_1 = 2 P_2$ , tem-se para o caudal na carga  $Q_L$ :

$$Q_L = Q_1 = \begin{cases} (\bar{x}_s + \bar{r}) \times cte \times \sqrt{\frac{P_s}{2}} + g_{lkm} \frac{P_s}{2} - (\bar{r} - \bar{x}_s) \times cte \times \sqrt{\frac{P_s}{2}} + g_{lkm} \frac{P_s}{2} & ; \bar{x}_s \leq \bar{r} \\ (\bar{x}_s + \bar{r}) \times cte \times \sqrt{\frac{P_s}{2}} + g_{lkm} \frac{P_s}{2} - (g_{lkm} e^{-k(\bar{x}_s - \bar{r})}) \frac{P_s}{2} & ; \bar{x}_s > \bar{r} \end{cases} \quad (5.24)$$



Simplificando as equações tem-se:

$$Q_L = \begin{cases} 2\bar{x}_s \times cte \times \sqrt{\frac{P_s}{2}} & ; \bar{x}_s \leq \bar{r} \\ (\bar{x}_s + \bar{r}) \times cte \times \sqrt{\frac{P_s}{2}} + g_{lkm} \frac{P_s}{2} (1 - e^{-k(\bar{x}_s - \bar{r})}) & ; \bar{x}_s > \bar{r} \end{cases} \quad (5.25)$$

e a derivada de  $Q_L$  em ordem a  $\bar{x}_s$

$$\frac{dQ_L}{d\bar{x}_s} = \begin{cases} 2 \times cte \times \sqrt{\frac{P_s}{2}} & ; \bar{x}_s \leq \bar{r} \\ cte \times \sqrt{\frac{P_s}{2}} + g_{lkm} \frac{P_s}{2} k \times e^{-k(\bar{x}_s - \bar{r})} & ; \bar{x}_s > \bar{r} \end{cases} \quad (5.26)$$

As equações para o caudal, em função da posição da gaveta da válvula, garantem a continuidade do caudal. Porém a derivada em ordem a  $\bar{x}_s$  não é contínua para  $\bar{x}_s = \bar{r}$ .

O ganho de caudal na origem é então:

$$\bar{K}_{q0} = 2 \times \frac{Q_n}{\sqrt{P_n}} \sqrt{\frac{P_s}{2}} \quad (5.27)$$

Neste caso não é possível ajustar o ganho de caudal.

### Ganho de pressão:

Para este tipo de modelos (componente de escoamento laminar e turbulento), o processo consiste na obtenção de uma expressão para a pressão na carga,  $P_L$ , a partir da expressão linearizada de  $Q_L$ . Este processo permite obter uma aproximação razoável para o ganho de pressão.

Neste caso utiliza-se a expressão para  $Q_L$  válida para a região de recobrimento negativo:

$$Q_L = Q_1 = q_{s1} - q_{1t} = (\bar{x}_s + \bar{r}) \times cte \times \sqrt{DP_{s1}} + g_{lkm} \times DP_{s1} - (\bar{r} - \bar{x}_s) \times cte \times \sqrt{DP_{1t}} - g_{lkm} \times DP_{1t} \quad ; \bar{x}_s < \bar{r} \quad (5.28)$$

Usando a equação (5.11) para exprimir  $Q_L$  em função da posição da gaveta e da pressão na carga obtém-se:

$$Q_L = (\bar{x}_s + \bar{r}) \times cte \times \sqrt{\frac{P_s}{2} (1 - \bar{P}_L)} + (\bar{x}_s - \bar{r}) \times cte \times \sqrt{\frac{P_s}{2} (1 + \bar{P}_L)} - g_{lkm} \times P_s \times \bar{P}_L \quad ; \bar{x}_s < \bar{r} \quad (5.29)$$

Linearizando em torno de um ponto  $(\bar{x}_{s0}, \bar{P}_{L0})$ , sito na região de recobrimento negativo, tem-se

$$Q_L = Q_L(\bar{x}_{s0}, \bar{P}_{L0}) + \frac{\partial Q_L}{\partial \bar{x}_s} \times cte \times \sqrt{\frac{P_s}{2}(1 - \bar{P}_L)} + cte \times \sqrt{\frac{P_s}{2}(1 + \bar{P}_L)} \frac{\partial Q_L}{\partial \bar{P}_L} (\bar{x}_s - \bar{x}_{s0}) + \frac{\partial^2 Q_L}{\partial \bar{x}_s^2} \frac{(\bar{x}_s + \bar{r}) \times cte}{2 \times \sqrt{(1 - \bar{P}_L)}} \sqrt{\frac{P_s}{2}} + \frac{(\bar{x}_s - \bar{r}) \times cte}{2 \times \sqrt{(1 + \bar{P}_L)}} \sqrt{\frac{P_s}{2}} - g_{lkm} \times P_s \frac{\partial^2 Q_L}{\partial \bar{P}_L^2} (\bar{P}_L - \bar{P}_{L0}) \quad (5.30)$$

A linearização em  $(\bar{x}_{s0}, \bar{P}_{L0}) = (0, 0)$  resulta para  $Q_L$ :

$$Q_L = 2\bar{x}_s \times cte \times \sqrt{\frac{P_s}{2}} - \frac{\partial Q_L}{\partial \bar{r}} \times cte \times \sqrt{\frac{P_s}{2}} + g_{lkm} \times P_s \frac{\partial^2 Q_L}{\partial \bar{P}_L^2} \quad (5.31)$$

A expressão para a pressão relativa na carga ( $\bar{P}_L$ ) pode obter-se anulando  $Q_L$  na equação (5.31):

$$\bar{P}_L = \frac{2\bar{x}_s \times cte \times \sqrt{\frac{P_s}{2}}}{\bar{r} \times cte \times \sqrt{\frac{P_s}{2}} + g_{lkm} \times P_s} \quad (5.32)$$

obtendo-se a seguinte expressão para a derivada da pressão relativa na carga (com  $cte = \frac{Q_n}{\sqrt{P_n}}$ ):

$$\frac{\partial \bar{P}_L}{\partial \bar{x}_s} = \frac{\frac{Q_n}{\sqrt{P_n}} \sqrt{2}}{\bar{r} \times \frac{Q_n}{\sqrt{P_n}} \sqrt{\frac{1}{2}} + g_{lkm} \times \sqrt{P_s}} \quad (5.33)$$

e o ganho relativo de pressão vem dado por:

$$\bar{K}_{p0} = \frac{Q_n \times \sqrt{2}}{\bar{r} \times Q_n \times \sqrt{\frac{1}{2}} + g_{lkm} \times \sqrt{P_n} \times P_s} \quad (5.34)$$

Se nos cálculos  $P_s = 2 P_n$ , o ganho de pressão relativo é dado por:

$$\bar{K}_{p0} = \frac{2 \times Q_n}{\bar{r} \times Q_n + g_{lkm} \times P_s} \quad (5.35)$$

### Caudal de fugas:

Partindo da equação (5.12), tendo em consideração (5.11), e dentro da região de *underlap*,  $|\bar{x}_s| < u$

$$q_{lk} = (\bar{r} - \bar{x}_s) \times cte \times \sqrt{\frac{P_s}{2}(1 + \bar{P}_L)} + g_{lkm} \times P_s + (\bar{x}_s + \bar{r}) \times cte \times \sqrt{\frac{P_s}{2}(1 - \bar{P}_L)} \quad ; \bar{x}_s < \bar{r} \quad (5.36)$$

Na posição central,  $\bar{x}_s = 0$ ;  $\bar{P}_L = 0$ , o caudal de fugas é:

$$q_{lk0} = \bar{r} \times \frac{Q_n}{\sqrt{P_n}} \times \sqrt{2 \times P_s} + g_{lkm} \times P_s \quad (5.37)$$

### Parametrização e simulações:

A partir de (5.27), (5.34) e (5.37) pode verificar-se que o parâmetro  $l$  não interfere nas características estáticas da válvula (na região intermédia), existindo somente dois parâmetros para ajuste:  $g_{lkm}$  e  $\bar{r}$ .  $l$  pode ser usado para definir o decaimento da curva de caudal de fugas fora da região de recobrimento negativo. Utilizando os dados da válvula de teste, o ajuste simultâneo do ganho de caudal e do ganho de pressão origina um valor negativo para  $\bar{r}$ , isto é, leva a válvula a ter um recobrimento positivo.

As figuras seguintes mostram os resultados resultantes da simulação com ajuste do ganho relativo de pressão, usando a equação (5.34) e fixando a condutância de fugas máxima,  $g_{lkm}$ .

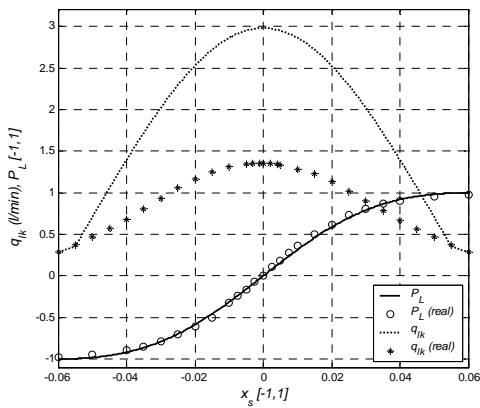


Figura 5.9 Pressão relativa na carga e caudal de fugas ( $\bar{r} = 0.055$ ,  $l = 40$ ,  $g_{lkm} = 4.2 \times 10^{-13}$ ) e  $\bar{x}_s \hat{I}$  [- 6%, 6% ]

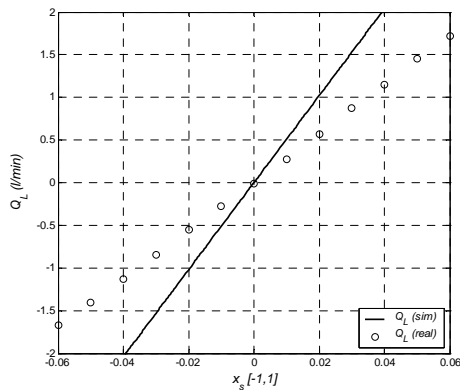


Figura 5.10 Caudal na carga para  $P_s = 2 P_n$  ( $\bar{r} = 0.055$ ,  $l = 40$ ,  $g_{lkm} = 4.2 \times 10^{-13}$ ) e  $\bar{x}_s \hat{I}$  [- 6%, 6% ]

## Modelo estático 2

Este modelo considera que as condutâncias de fugas ( $g_{lkp}$ ,  $g_{lkn}$ ) versus posição relativa da gaveta ( $\bar{x}_s \in [-1, 1]$ ) são modeladas por função exponenciais:

$$g_{lkp} = \begin{cases} g_{lkm} e^{l\bar{x}_s} & ; \bar{x}_s < 0 \\ g_{lkm} & ; \bar{x}_s \geq 0 \end{cases}$$

$$g_{lkn} = \begin{cases} g_{lkm} e^{-l\bar{x}_s} & ; \bar{x}_s > 0 \\ g_{lkm} & ; \bar{x}_s \leq 0 \end{cases} \quad (5.38)$$

$g_{lkm}$  é a condutância de fugas máxima,  $l$  define o decaimento da curva exponencial.

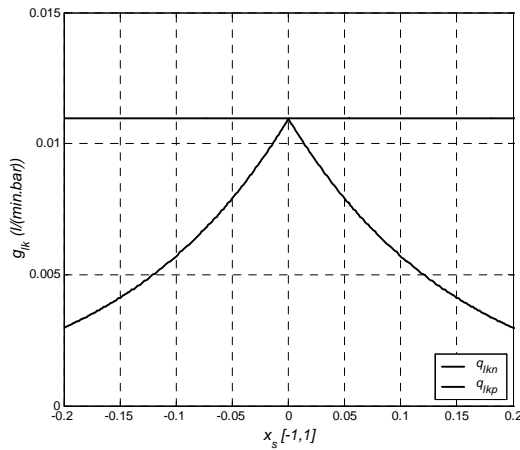


Figura 5.11 Característica da condutância de fugas em função da posição relativa da gaveta

Considera-se que a condutância de fugas da válvula é máxima quando o orifício não se encontra sobreposto. Quando o orifício está sobreposto, a condutância é modelada por uma função baseada na função exponencial onde a constante  $l$  define o decaimento da exponencial.

Neste caso as pseudo-seções são dadas por:

$$A_{s1} = A_{2t} = A_p = \begin{cases} \bar{x}_s \times cte & ; \bar{x}_s \geq 0 \\ 0 & ; \bar{x}_s < 0 \end{cases}$$

$$A_{s2} = A_{1t} = A_n = \begin{cases} -\bar{x}_s \times cte & ; \bar{x}_s < 0 \\ 0 & ; \bar{x}_s \geq 0 \end{cases} \quad (5.39)$$

$$\text{com } cte = \frac{Q_n}{\sqrt{P_n}}$$

### Ganho de caudal:

Analisando para  $\bar{x}_s > 0$ , e considerando  $P_s = 2 P_n = 2 P_1 = 2 P_2$  tem-se, para o caudal na carga,  $Q_L$ :

$$Q_L = Q_1 = \bar{x}_s \times cte \times \sqrt{\frac{P_s}{2}} + g_{lkm} \times \frac{P_s}{2} - g_{lkm} \times e^{-l\bar{x}_s} \times \frac{P_s}{2} \quad (5.40)$$

Então o ganho de caudal na posição intermédia vem dado por:

$$\left. \frac{\partial Q_L}{\partial \bar{x}_s} \right|_{\bar{x}_s=0} = \bar{K}_{q0} = \frac{Q_n}{\sqrt{P_n}} \sqrt{\frac{P_s}{2}} + g_{lkm} \times l \times \frac{P_s}{2} \quad (5.41)$$

### Ganho de pressão:

O processo de obtenção da expressão para o ganho de pressão é idêntico ao usado para o modelo estático 1. A partir da expressão para  $Q_L$  em função de  $\bar{x}_s$  e  $\bar{P}_L$ :

$$Q_L = \bar{x}_s \times cte \times \sqrt{\frac{P_s}{2}(1 - \bar{P}_L)} + g_{lkm} \times \frac{P_s}{2}(1 - \bar{P}_L) - g_{lkm} \times e^{-l\bar{x}_s} \times \frac{P_s}{2}(1 + \bar{P}_L) \quad (5.42)$$

Linearizando  $Q_L$  na posição  $(\bar{x}_{s0}, \bar{P}_{L0})$  tem-se:

$$Q_L = Q_L(\bar{x}_{s0}, \bar{P}_{L0}) + \left. \frac{\partial Q_L}{\partial \bar{x}_s} \right|_{\bar{x}_{s0}, \bar{P}_{L0}} (\bar{x}_s - \bar{x}_{s0}) + \left. \frac{\partial Q_L}{\partial \bar{P}_L} \right|_{\bar{x}_{s0}, \bar{P}_{L0}} (\bar{P}_L - \bar{P}_{L0}) \quad (5.43)$$

$$\left. \frac{\partial Q_L}{\partial \bar{x}_s} \right|_{\bar{x}_{s0}, \bar{P}_{L0}} = cte \times \sqrt{\frac{P_s}{2}(1 - \bar{P}_{L0})} - g_{lkm} \times e^{-l\bar{x}_{s0}} \times \frac{P_s}{2} (1 + \bar{P}_{L0})$$

$$\left. \frac{\partial Q_L}{\partial \bar{P}_L} \right|_{\bar{x}_{s0}, \bar{P}_{L0}} = \frac{\bar{x}_{s0} \times cte}{\sqrt{2 \times P_s}} \sqrt{1 - \bar{P}_{L0}} - g_{lkm} \times \frac{P_s}{2} - g_{lkm} \times e^{-l\bar{x}_{s0}} \times \frac{P_s}{2} (\bar{P}_{L0} - 1)$$

Da linearização em  $(\bar{x}_{s0}, \bar{P}_{L0}) = (0, 0)$  resulta para  $Q_L$ :

$$Q_L = \bar{x}_s \times cte \times \sqrt{\frac{P_s}{2}} + g_{lkm} \times l \times \frac{P_s}{2} \bar{x}_s - g_{lkm} \times P_s \times \bar{P}_L \quad (5.44)$$

Para calcular a pressão relativa na carga  $(\bar{P}_L)$  anula-se  $Q_L$

$$\bar{P}_L = \frac{\bar{x}_s}{g_{lkm}} \frac{Q_n}{\sqrt{P_n}} \sqrt{\frac{1}{2 \times P_s}} + \frac{l \times \bar{x}_s}{2} \quad (5.45)$$

derivando (5.45) em ordem a  $\bar{x}_s$  tem-se para o ganho relativo de pressão:

$$\bar{K}_{p0} = \left. \frac{\partial \bar{P}_L}{\partial \bar{x}_s} \right|_{\bar{x}_s=0} = \frac{Q_n}{g_{lkm} \sqrt{P_n}} \sqrt{\frac{1}{2 \times P_s}} + \frac{l}{2} \quad (5.46)$$

### Caudal de fugas:

Partindo de (5.12), analisando para  $\bar{x}_s > 0$  e  $P_t = 0$ , e exprimindo em função de  $\bar{P}_L$  através de (5.11), tem-se, para o caudal de fugas:

$$q_{lk} = g_{lk}(\bar{x}_s, \bar{P}_L) = q_{1t} + q_{2t} = (g_{lkm} \times e^{-l\bar{x}_s}) \frac{P_s}{2} (1 + \bar{P}_L) + \bar{x}_s \times cte \times \sqrt{\frac{P_s}{2} (1 - \bar{P}_L)} + g_{lkm} \times \frac{P_s}{2} (1 - \bar{P}_L) \quad (5.47)$$

Na posição central,  $\bar{x}_s = 0$ ;  $\bar{P}_L = 0$ , o caudal de fugas é então:

$$q_{lk0} = g_{lkm} \times P_s \quad (5.48)$$

### Parametrização e simulações:

Existem duas possibilidades de cálculo dos parâmetros do modelo (assumindo que o ganho relativo de pressão é uma característica sempre usada na parametrização): a “reprodução” do caudal de fugas na origem ou do ganho de caudal.

A reprodução do caudal de fugas máximo na origem pressupõe a utilização de (5.48) e de (5.46), sendo os parâmetros extraídos de forma algébrica através das relações seguintes:

$$g_{lkm} = \frac{q_{lk0}}{P_s}; \quad l = 2\bar{K}_{p0} - \frac{Q_n}{g_{lkm} \sqrt{P_n}} \sqrt{\frac{2}{P_s}} \quad (5.49)$$

Os resultados comparativos são apresentados nas figuras seguintes para:  $g_{lkm} = 0.0194 \frac{l}{\text{min} \times \text{bar}}$  e  $l = 35.5$ .

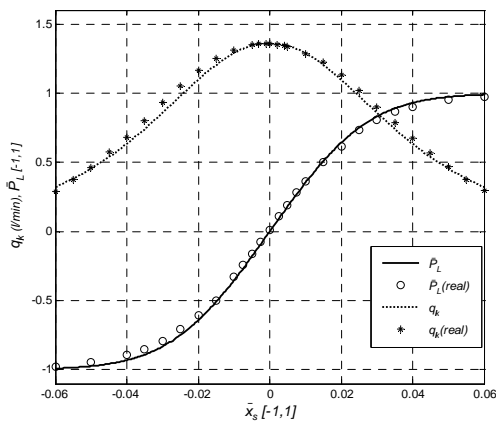


Figura 5.12 Pressão relativa na carga e caudal de fugas ( $l = 35.5$ ,  $g_{lkm} = 0.0194 \frac{l}{\text{min} \times \text{bar}}$ ) e e  $\bar{x}_s \hat{I}$  [- 6%, 6% ]

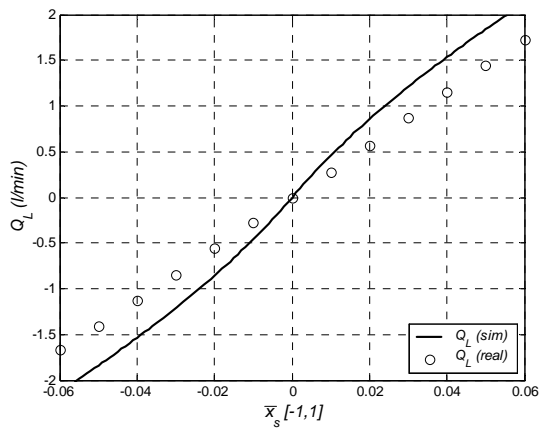


Figura 5.13 Caudal na carga ( $l=35.5$ ,  $g_{lkm} = 0.0194 \frac{l}{\text{min} \times \text{bar}}$ ) e  $\bar{x}_s \hat{I}$  [- 6%, 6% ]

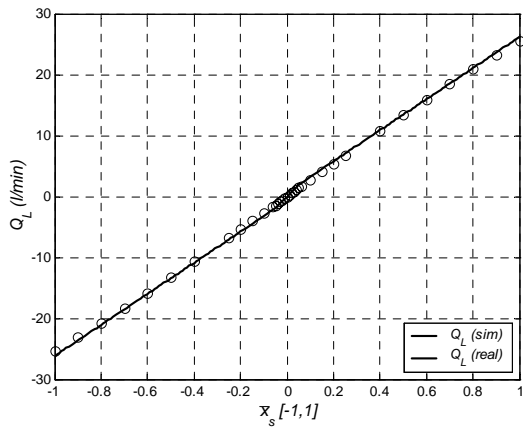


Figura 5.14 Caudal na carga ( $l=35.5$ ,  $g_{lkm} = 0.0194 \frac{l}{\text{min} \times \text{bar}}$ ) e  $\bar{x}_s \hat{I}$  [- 1,1 ]

A reprodução do ganho de caudal pressupõe a utilização de (5.41) e de (5.46), sendo os parâmetros extraídos através do seguinte sistema de equações não lineares:

$$\begin{aligned} \bar{K}_{p0} &= \frac{Q_n}{g_{lkm} \sqrt{P_n}} \sqrt{\frac{1}{2 \times P_s}} + \frac{l}{2} \\ \bar{K}_{q0} &= \frac{Q_n}{\sqrt{P_n}} \sqrt{\frac{P_s}{2}} + g_{lkm} \times l \times \frac{P_s}{2} \end{aligned} \quad (5.50)$$

Tem-se então os seguintes valores para os parâmetros:  $g_{lkm} = 0.011 \frac{l}{\text{min} \times \text{bar}}$  e  $l = 6.52$

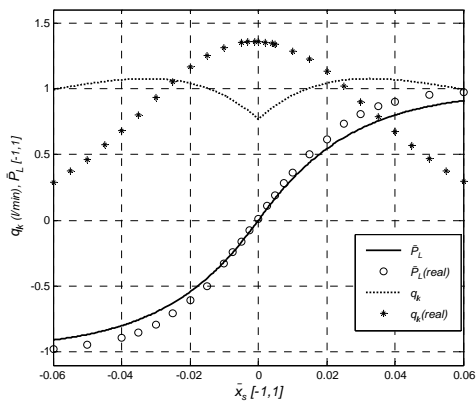


Figura 5.15 Pressão relativa na carga e caudal de fugas ( $l = 6.52$ ,  $g_{lkm} = 0.0110 \frac{l}{\text{min} \times \text{bar}}$ ) e  $\bar{x}_s \hat{I} [-6\%, 6\%]$

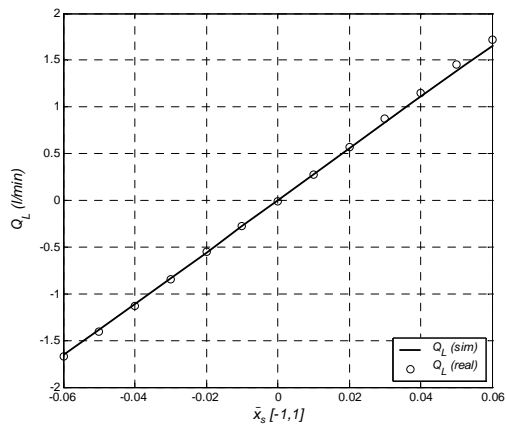


Figura 5.16 Caudal na carga ( $l = 6.52$ ,  $g_{lkm} = 0.0110 \frac{l}{\text{min} \times \text{bar}}$ ) e  $\bar{x}_s \hat{I} [-6\%, 6\%]$

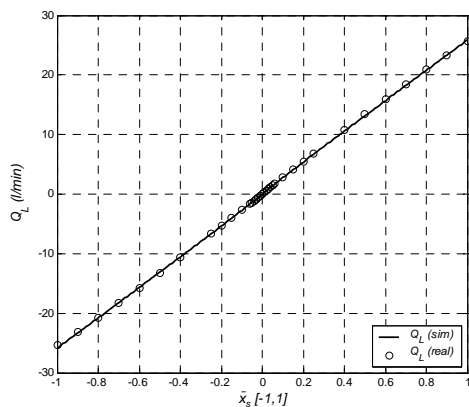


Figura 5.17 Caudal na carga ( $l = 6.52$ ,  $g_{lkm} = 0.0110 \frac{l}{\text{min} \times \text{bar}}$ ) e  $\bar{x}_s \hat{I} [-1, 1]$



Quando o ganho de caudal é menor (em termos numéricos) que o caudal nominal, o modelo estático 2 não pode ser usado pois a sua parametrização resulta em valores de  $l$  negativos. Pode-se, no entanto, usar o modelo ajustando o caudal de fugas máximo. Note-se também que quanto maior for o ganho de caudal (válvulas com centro aberto), mais a forma do caudal de fugas se assemelha à forma real, e mais a curva do ganho de pressão se aproxima da curva real, como se pode ver na Figura 5.18 onde se simulou com  $\overline{K}_{q0} = 40$ .

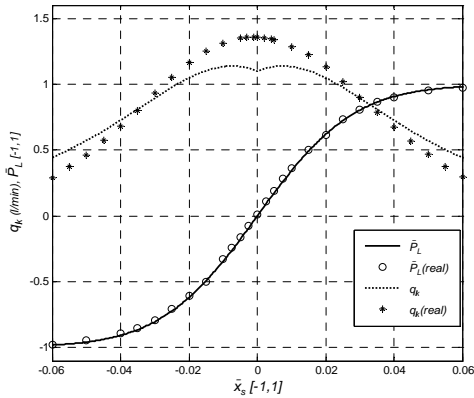


Figura 5.18 Pressão relativa na carga e caudal de fugas ( $l = 26.5$ ,  $g_{lkm} = 0.0157 \frac{l}{\text{min} \times \text{bar}}$ ) e  $\bar{x}_s \hat{I} [-6\%, 6\%]$

### 5.1.3 Modelos com escoamento turbulento

Outro tipo de modelos propostos, por exemplo em [Linjama98] e [Quintas99], consideram o escoamento nos orifícios como sendo sempre turbulentos, modelando-se os escoamentos laminares implicitamente nas pseudo-seções  $A_{ij}(\bar{x}_s)$ . Na Figura 5.19 mostra-se o traçado típico destas funções para uma válvula simétrica e emparelhada.

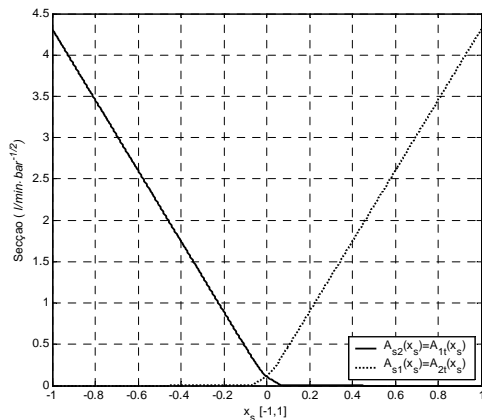


Figura 5.19 Pseudo-seções em função da posição da gaveta

Usando (5.4), e anulando as condutâncias de fugas, obtêm-se as equações genéricas para os caudais através da válvula:

$$\begin{aligned} q_{s1} &= \text{sign}(DP_{s1}) \times A_{s1}(\bar{x}_s) \times \sqrt{|DP_{s1}|} \\ q_{s2} &= \text{sign}(DP_{s2}) \times A_{s2}(\bar{x}_s) \times \sqrt{|DP_{s2}|} \\ q_{1t} &= \text{sign}(DP_{1t}) \times A_{1t}(\bar{x}_s) \times \sqrt{|DP_{1t}|} \\ q_{2t} &= \text{sign}(DP_{2t}) \times A_{2t}(\bar{x}_s) \times \sqrt{|DP_{2t}|} \end{aligned} \quad (5.51)$$

Para uma válvula simétrica e centrada, com os orifícios da gaveta e do corpo da válvula completamente emparelhados (caso mais usual), o número de pseudo-seções pode ser reduzido. Desta forma, assumindo emparelhamento e simetria, as equações para os caudais através da válvula (5.51) podem ser simplificadas:

$$\begin{aligned} q_{s1} &= \text{sign}(DP_{s1}) \times A_p(\bar{x}_s) \times \sqrt{|DP_{s1}|} \\ q_{s2} &= \text{sign}(DP_{s2}) \times A_n(\bar{x}_s) \times \sqrt{|DP_{s2}|} \\ q_{1t} &= \text{sign}(DP_{1t}) \times A_n(\bar{x}_s) \times \sqrt{|DP_{1t}|} \\ q_{2t} &= \text{sign}(DP_{2t}) \times A_p(\bar{x}_s) \times \sqrt{|DP_{2t}|} \end{aligned} \quad (5.52)$$

Nas próximas seções são apresentadas as expressões para o ganho de caudal, o ganho de pressão e o caudal de fugas de uma válvula centrada.

### Ganho de caudal

Considerando o diagrama da Figura 5.3, se os terminais 1 e 2 forem ligados por uma resistência nula e, para simplificar a notação, assumindo  $\Delta P_{ij}$  sempre positivo, *i.e.*  $\text{sign}(DP_{ij}) = 1$ , o caudal na carga,  $Q_L$ , resulta em:

$$Q_L = q_{s1} - q_{1t} = A_p(\bar{x}_s) \times \sqrt{DP_{s1}} - A_n(\bar{x}_s) \times \sqrt{DP_{1t}} \quad (5.53)$$

Assumindo  $P_t = 0$ , ou considerando  $P_t$  uma pressão de referência, o caudal na carga vem:

$$Q_L = A_p(\bar{x}_s) \times \sqrt{P_s - P_1} - A_n(\bar{x}_s) \times \sqrt{P_1} \quad (5.54)$$

onde  $P_i$  é a pressão relativa no terminal  $i$  referido à pressão  $P_t$  de referência.

Como os terminais 1 e 2 estão ligados com uma resistência nula,

$$P_1 = P_2 = \frac{P_s}{2}, \quad " \quad Q_L(\bar{x}_s) \quad (5.55)$$

então o caudal na carga vem dado por:

$$Q_L = A_p (\bar{x}_s) \times \sqrt{\frac{P_s}{2}} - A_n (\bar{x}_s) \times \sqrt{\frac{P_s}{2}} \quad (5.56)$$

e o ganho de caudal pode obter-se por diferenciação da equação (5.56).

### Ganho de pressão

No diagrama da válvula da Figura 5.6, os caudais internos são iguais dois a dois ( $q_{1t} = q_{s1}$  e  $q_{2t} = q_{s2}$ ). Então (5.11) e (5.53) conduzem à seguinte relação:

$$A_p (\bar{x}_s) \times \sqrt{\frac{P_s}{2} (1 - \bar{P}_L)} = A_n (\bar{x}_s) \times \sqrt{\frac{P_s}{2} (1 + \bar{P}_L)} \quad (5.57)$$

Consequentemente, resolvendo (5.57) em ordem a  $\bar{P}_L$ , a pressão relativa na carga é dada por:

$$\bar{P}_L = \frac{A_p (\bar{x}_s)^2 - A_n (\bar{x}_s)^2}{A_p (\bar{x}_s)^2 + A_n (\bar{x}_s)^2} \quad (5.58)$$

### Caudal de fugas

A partir de (5.12) e (5.52), e analisando  $q_{lk}$  para  $\bar{x}_s \neq 0$ :

$$q_{lk} = A_n (\bar{x}_s) \times \sqrt{P_1} + A_n (\bar{x}_s) \times \sqrt{P_s - P_2} = A_p (\bar{x}_s) \times \sqrt{P_s - P_1} + A_p (\bar{x}_s) \times \sqrt{P_2} \quad (5.59)$$

Usando (5.11) e (5.59), o caudal de fugas pode ser expresso como uma função da pressão relativa na carga,  $\bar{P}_L$ :

$$q_{lk} = q_{lk} (\bar{x}_s, \bar{P}_L) = 2A_n (\bar{x}_s) \times \sqrt{\frac{P_s}{2} (1 + \bar{P}_L)} \quad (5.60)$$

De forma idêntica, o caudal de fugas pode ser expresso como uma função da pseudo-secção  $A_p (\bar{x}_s)$ :

$$q_{lk} = q_{lk} (\bar{x}_s, \bar{P}_L) = 2A_p (\bar{x}_s) \times \sqrt{\frac{P_s}{2} (1 - \bar{P}_L)} \quad (5.61)$$

Na posição central, *i.e.*  $\bar{x}_s = 0$ ,  $\bar{P}_L = 0$ , o caudal de fugas vem dado por:

$$q_{lk} (\bar{x}_s, \bar{P}_L) \Big|_{\bar{x}_s = 0} = q_{lk0} = 2 \times A_n (0) \times \sqrt{\frac{P_s}{2}} \quad (5.62)$$

O caudal de fugas numa posição positiva da gaveta,  $\bar{x}_s^*$ , distante da posição central, pode ser definido assumindo que  $\bar{P}_L = 1$  em  $\bar{x}_s^*$ . O uso deste pressuposto em (5.60) origina a seguinte equação para  $q_{lk}$ :

$$q_{lk} = 2 \times A_n (\bar{x}_s^*) \times \sqrt{\bar{P}_s} \quad (5.63)$$

São propostas diferentes soluções para a construção das funções pseudo-secção. Uma das aproximações utiliza equações com estrutura variável para a caracterização das pseudo-secções. A outra aproximação usa funções com estrutura fixa baseadas na função hiperbólica. Note-se que modelando somente o escoamento turbulento, e de acordo com (5.18), somente é possível ajustar dois dos três parâmetros estáticos da válvula.

### Modelo estático 3 (estrutura variável)

O modelo mais simples aqui apresentado para a pseudo-secção variável, baseia-se no modelo proposto em [Linjama98]. Este modelo utiliza o parâmetro  $\bar{u}_l$  para representar a abertura relativa da gaveta da válvula em  $\bar{x}_s = 0$  e as seguintes expressões para as pseudo-secções:

$$A_p(\bar{x}_s) = \begin{cases} \frac{Q_n}{\sqrt{P_n}} \bar{x}_s & ; e \times \bar{u}_l \leq \bar{x}_s \leq 1 \\ \frac{Q_n}{\sqrt{P_n}} \bar{u}_l \times e^{\frac{\bar{x}_s}{\bar{u}_l}} & ; -1 \leq \bar{x}_s < e \times \bar{u}_l \end{cases}$$

$$A_n(\bar{x}_s) = \begin{cases} -\frac{Q_n}{\sqrt{P_n}} \bar{x}_s & ; -1 \leq \bar{x}_s \leq -e \times \bar{u}_l \\ \frac{Q_n}{\sqrt{P_n}} \bar{u}_l \times e^{\frac{-\bar{x}_s}{\bar{u}_l}} & ; -e \times \bar{u}_l < \bar{x}_s \leq 1 \end{cases} \quad (5.64)$$

com  $\bar{u}_l \in ]0, 1[$ .

As equações usadas para o cálculo dos parâmetros das pseudo-secções são apresentadas a seguir. De (5.58) e (5.64), a pressão relativa na carga perto da posição intermédia é dada por:

$$\bar{P}_L = \frac{e^{\frac{2\bar{x}_s}{\bar{u}_l}} - e^{-\frac{2\bar{x}_s}{\bar{u}_l}}}{e^{\frac{2\bar{x}_s}{\bar{u}_l}} + e^{-\frac{2\bar{x}_s}{\bar{u}_l}}} = \tanh\left(\frac{2 \times \bar{x}_s}{\bar{u}_l}\right); \quad -\bar{u}_l \leq \bar{x}_s \leq \bar{u}_l \quad (5.65)$$

Então, o ganho de pressão relativa (5.10) na origem é:

$$\bar{K}_{p0} = \frac{2}{e \times \bar{u}_l} \quad (5.66)$$

De (5.58) e (5.61) o caudal de fugas pode ser expresso por:

$$q_{lk} = 2A_n(\bar{x}_s) \times A_p(\bar{x}_s) \sqrt{\frac{P_s}{(A_p(\bar{x}_s))^2 + A_n(\bar{x}_s)^2}} \quad (5.67)$$

Para  $-\bar{u}_l \leq \bar{x}_s \leq \bar{u}_l$  as pseudo-seccões são  $A_p(\bar{x}_s) = \frac{Q_n}{\sqrt{P_n}} \bar{u}_l \times e^{\frac{\bar{x}_s}{e \times \bar{u}_l}}$  e  $A_n(\bar{x}_s) = \bar{u}_l \times \frac{Q_n}{\sqrt{P_n}} e^{-\frac{\bar{x}_s}{e \times \bar{u}_l}}$ .

Quando estas são substituídas em (5.67), o caudal de fugas vem dado por:

$$q_{lk} = 2 \times \bar{u}_l^2 \times \frac{Q_n^2}{P_n} \sqrt{\frac{P_s}{\bar{u}_l^2 \frac{Q_n^2}{P_n} e^{\frac{2\bar{x}_s}{e \times \bar{u}_l}} + \bar{u}_l^2 \frac{Q_n^2}{P_n} e^{-\frac{2\bar{x}_s}{e \times \bar{u}_l}}}} = \bar{u}_l \frac{Q_n}{\sqrt{P_n}} \sqrt{\frac{2 \times P_s}{\cosh\left(\frac{2 \times \bar{x}_s}{e \times \bar{u}_l}\right)}} \quad (5.68)$$

Note-se que a função  $1/\cosh(\cdot)$  tem uma evolução semelhante à curva real típica do caudal de fugas.

O caudal de fugas máximo ocorre na posição intermédia e também depende de  $\bar{u}_l$ :

$$q_{lk0} = \bar{u}_l \frac{Q_n}{\sqrt{P_n}} \sqrt{2P_s} \quad (5.69)$$

Da equação (5.56), e para  $-\bar{u}_l \leq \bar{x}_s \leq \bar{u}_l$ , o caudal na carga é dado por:

$$Q_L = \bar{u}_l \times \frac{Q_n}{\sqrt{P_n}} e^{\frac{\bar{x}_s}{e \times \bar{u}_l}} \times \sqrt{\frac{P_s}{2}} - \bar{u}_l \times \frac{Q_n}{\sqrt{P_n}} e^{-\frac{\bar{x}_s}{e \times \bar{u}_l}} \times \sqrt{\frac{P_s}{2}} \quad (5.70)$$

Então, o ganho de caudal é:

$$\bar{K}_{q0} = \frac{Q_n}{e \times \sqrt{P_n}} \times \sqrt{2 \times P_s} \quad (5.71)$$

Pode-se concluir que, com esta estrutura para a pseudo-seccão, só é possível ajustar o ganho relativo de pressão ou o caudal de fugas máximo (não os dois simultaneamente), pois estas características estáticas dependem de um só parâmetro do modelo,  $u_l$ . O ganho de caudal não pode ser ajustado pois só depende do caudal nominal,  $Q_n$ , que é usado para ajustar o caudal na abertura máxima da válvula (para uma diferença de pressão no orifício igual à pressão nominal  $P_n$ ).

O ajuste do ganho de pressão é realizado extraindo o parâmetro  $u_l$  da equação (5.66). Os resultados são apresentados nas figuras 5.20, 5.21 e 5.22.

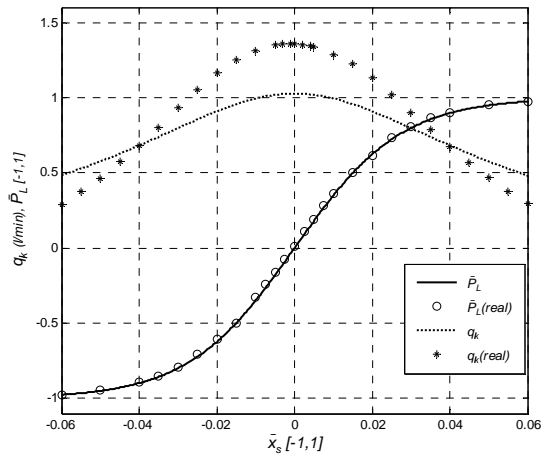


Figura 5.20 Pressão relativa na carga e caudal de fugas ( $\bar{u}_l = 0.0202$ ) e  $\bar{x}_s \hat{I}$  [- 6%, 6% ]

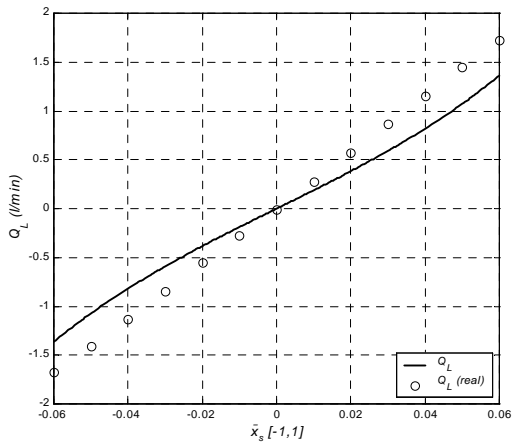


Figura 5.21 Caudal na carga ( $\bar{u}_l = 0.0202$ ) e  $\bar{x}_s \hat{I}$  [- 6%, 6% ]

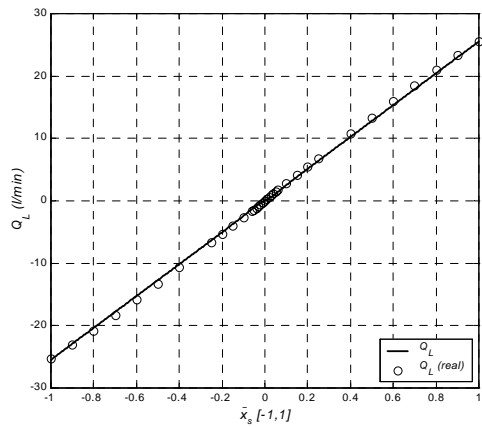


Figura 5.22 Caudal na carga ( $\bar{u}_l = 0.0202$ ) e  $\bar{x}_s \hat{I}$  [- 1, 1 ]

Para o ajuste do caudal de fugas na origem deve usar-se a equação (5.69). Neste caso, tem-se para o parâmetro  $\bar{u}_l$  :  $\bar{u}_l = 0.0267$  . Os resultados são apresentados nas figuras 5.23, 5.24 e 5.25.

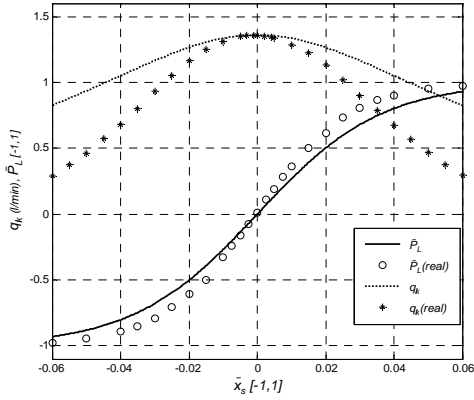


Figura 5.23 Pressão relativa na carga e caudal de fugas ( $\bar{u}_l = 0.0267$ ) e  $\bar{x}_s \hat{I} [-6\%, 6\%]$

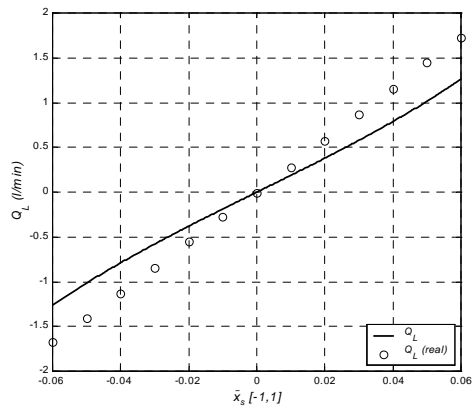


Figura 5.24 Caudal na carga ( $\bar{u}_l = 0.0267$ ) e  $\bar{x}_s \hat{I} [-6\%, 6\%]$

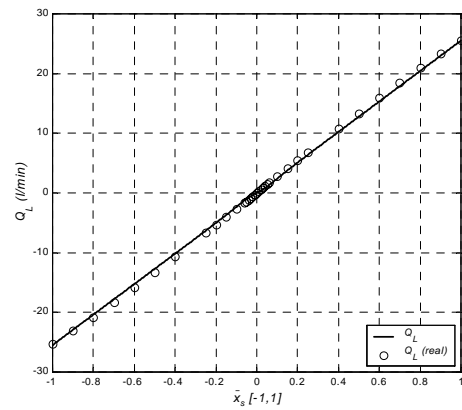


Figura 5.25 Caudal na carga ( $\bar{u}_l = 0.0267$ ) e  $\bar{x}_s \hat{I} [-1, 1]$

#### Modelo estático 4 (estrutura variável)

Este modelo para as pseudo-seções foi proposto em [Quintas99]. A característica principal desta aproximação é a relação algébrica que resulta entre os parâmetros do modelo e as características estáticas da válvula.

Neste caso, as funções pseudo-seção têm duas regiões diferentes que transitam num ponto,  $\bar{x}_t$  :

$$A_n(\bar{x}_s) = \begin{cases} g \times e^{-\frac{k\bar{x}_s}{2}} & ; -\bar{x}_t \leq \bar{x}_s \leq 1 \\ -a \times \bar{x}_s + b & ; -1 \leq \bar{x}_s < -\bar{x}_t \end{cases} ;$$

$$A_p(\bar{x}_s) = \begin{cases} a \times \bar{x}_s + b & ; \bar{x}_t < \bar{x}_s \leq 1 \\ g \times e^{\frac{k\bar{x}_s}{2}} & ; -1 \leq \bar{x}_s \leq \bar{x}_t \end{cases} \quad (5.72)$$

com  $g, k, a, b, \bar{x}_t \in \mathbb{R}^+$  e  $\bar{x}_t \in ]0, 1[$ .

As características estáticas usadas para o cálculo dos parâmetros são: o caudal nominal ( $Q_n$ ) para uma diferença de pressão igual à pressão nominal  $P_n$ ; ganho de pressão relativa; caudal de fugas ou o ganho de caudal na posição intermédia.

As equações que podem ser usadas para o cálculo dos parâmetros das pseudo-seções são apresentadas a seguir. De (5.58) e (5.72), a pressão relativa na carga perto da posição intermédia é dada por:

$$\bar{P}_L = \frac{e^{k\bar{x}_s} - e^{-k\bar{x}_s}}{e^{k\bar{x}_s} + e^{-k\bar{x}_s}} = \tanh(k \times \bar{x}_s); \quad -\bar{x}_t \leq \bar{x}_s \leq \bar{x}_t \quad (5.73)$$

Então, o ganho de pressão relativa (5.10) na origem é:

$$\bar{K}_{p0} = k \quad (5.74)$$

De (5.58) e (5.61), o caudal de fugas pode ser expresso por:

$$q_{lk} = 2A_n(\bar{x}_s) \times A_p(\bar{x}_s) \sqrt{\frac{P_s}{(A_p(\bar{x}_s))^2 + (A_n(\bar{x}_s))^2}} \quad (5.75)$$

Para  $-\bar{x}_t \leq \bar{x}_s \leq \bar{x}_t$ , as pseudo-seções são  $A_p(\bar{x}_s) = g \times e^{\frac{k\bar{x}_s}{2}}$  e  $A_n(\bar{x}_s) = g \times e^{-\frac{k\bar{x}_s}{2}}$ . Quando estas são substituídas em (5.75), o caudal de fugas vem:

$$q_{lk} = 2g^2 \sqrt{\frac{P_s}{(g^2 e^{k\bar{x}_s} + g^2 e^{-k\bar{x}_s})}} = g \sqrt{\frac{2P_s}{\cosh(k \times \bar{x}_s)}} \quad (5.76)$$



O caudal de fugas na origem está directamente relacionado com o parâmetro  $\gamma$ :

$$q_{lk0} = g\sqrt{2P_s} \quad (5.77)$$

Da equação do caudal na carga (5.56), e para  $\bar{x}_t$  e  $\bar{x}_s$ , o caudal na carga é dado por:

$$Q_L = g \times e^{\frac{k\bar{x}_s}{2}} \times \sqrt{\frac{P_s}{2}} - g \times e^{\frac{-k\bar{x}_s}{2}} \times \sqrt{\frac{P_s}{2}} \quad (5.78)$$

Então, o ganho de caudal é:

$$\bar{K}_{q0} = k \times g \times \sqrt{\frac{P_s}{2}} \quad (5.79)$$

As outras equações usadas para o cálculo dos parâmetros são baseadas na continuidade da pseudo-secção  $A_p(\bar{x}_s)$  e da sua derivada, num determinado ponto de transição ( $\bar{x}_t$ ), e na definição de caudal nominal. O sistema de equações (5.80) deve ser resolvido para os parâmetros  $\alpha$ ,  $\beta$  e  $\bar{x}_t$ . O caudal nominal ( $Q_n$ ) à pressão nominal ( $P_n$ ), que está disponível no catálogo dos fabricantes ou pode ser medido para uma válvula específica, define o caudal “longe” da origem, pois  $A_n(\bar{x}_s) \gg 0$  para  $\bar{x}_s = 1$ .

$$\begin{cases} a \times \bar{x}_t + b = g \times e^{\frac{k\bar{x}_t}{2}} & \text{continuidade em } \bar{x}_t \\ a = g \times \frac{k}{2} \times e^{\frac{k\bar{x}_t}{2}} & \text{continuidade da derivada em } \bar{x}_t \\ Q_n|_{\bar{x}_s=1} = (a \times \bar{x}_s + b) \sqrt{P_n}|_{\bar{x}_s=1} & \text{definição de caudal nominal} \end{cases} \quad (5.80)$$

Uma propriedade interessante das pseudo-secções dadas em (5.72) é a relação directa entre os seus parâmetros e as características estáticas da válvula, nomeadamente o ganho de pressão ( $k$ ) e o caudal de fugas máximo ( $\gamma$ ). Assumindo que a pressão relativa deve ser sempre reproduzida, os parâmetros do modelo podem ser calculados de (5.74), (5.80) e (5.77) ou (5.79), para a reprodução na origem do caudal de fugas ou do ganho de caudal, respectivamente.

Usando a aproximação da estrutura variável para as pseudo-secções e seleccionando o ajuste do ganho de pressão e do caudal de fugas máximo, isto é, usando (5.74), (5.77) e (5.80), os valores para os parâmetros são:

$$k = 36.5, g = 0.1149m^4s^{-1}\sqrt{N}, a = 4.241m^4s^{-1}\sqrt{N}, b = 6.877 \times 10^{-2}m^4s^{-1}\sqrt{N}, \bar{x}_t = 3.858 \times 10^{-2}.$$

As figuras 5.26, 5.27 e 5.28 apresentam os resultados da simulação.

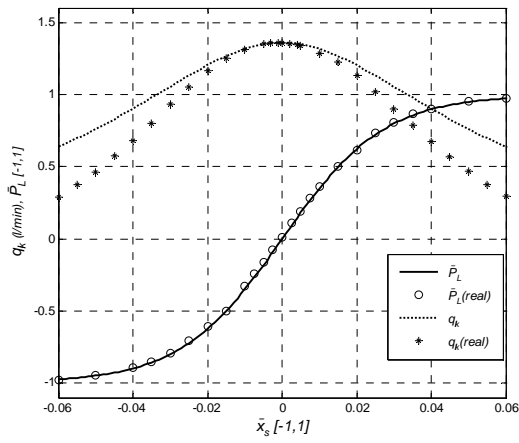


Figura 5.26 Caudal de fugas e ganho de pressão perto da posição intermédia:  $\bar{x}_s \hat{I}$  [- 6%, 6% ]

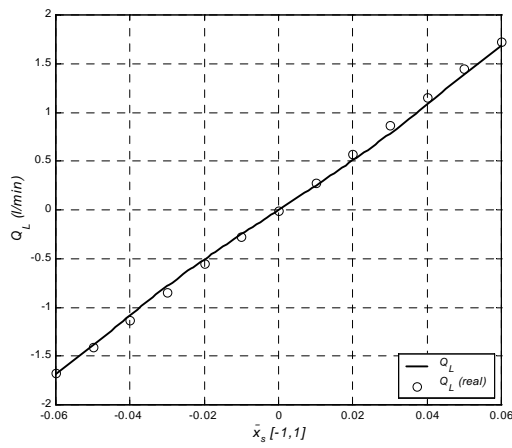


Figura 5.27 Caudal na carga perto da posição intermédia:  $\bar{x}_s \hat{I}$  [- 6%, 6% ]

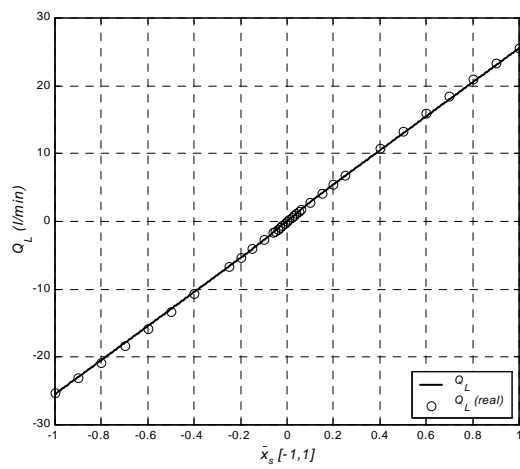


Figura 5.28 Caudal na carga para  $\bar{x}_s \hat{I}$  [- 100%, 100% ]

Usando a aproximação da estrutura variável para as pseudo-seções e seleccionando o ajuste do ganho de pressão e do ganho de caudal, isto é, usando (5.74), (5.79) e (5.80), os valores obtidos para os parâmetros do modelo são:

$$k = 36.5, g = 0.1297m^4s^{-1}\sqrt{N}, a = 4.387m^4s^{-1}\sqrt{N}, b = 9.199 \times 10^{-2}m^4s^{-1}\sqrt{N}, \bar{x}_t = 3.383 \times 10^{-2}.$$

Nas figuras 5.29, 5.30 e 5.31 ilustram-se os resultados da simulação, com os parâmetros anteriores, do modelo de estrutura variável.

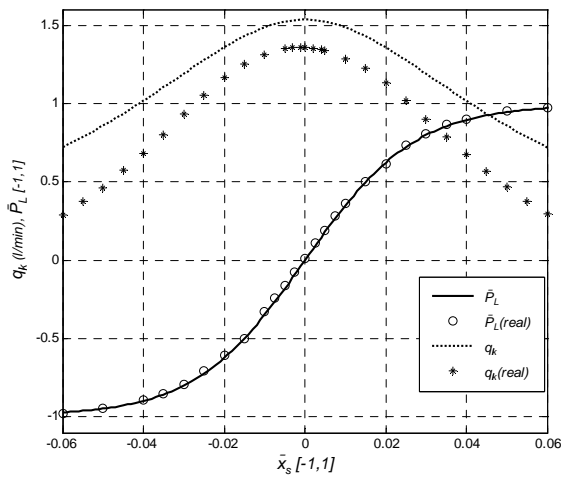


Figura 5.29 Caudal de fugas e ganho de pressão perto da posição intermédia:  $\bar{x}_s \hat{I} [-6\%, 6\%]$

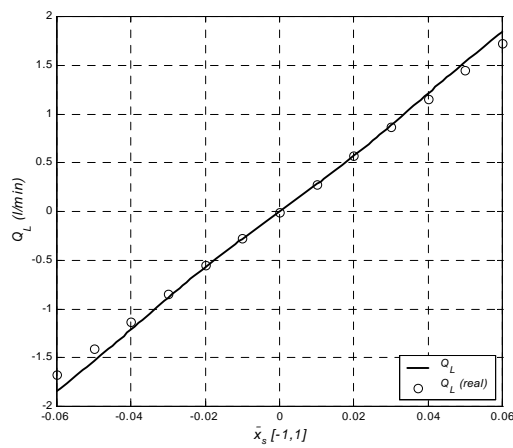


Figura 5.30 Caudal na carga na região intermédia:  $\bar{x}_s \hat{I} [-6\%, 6\%]$

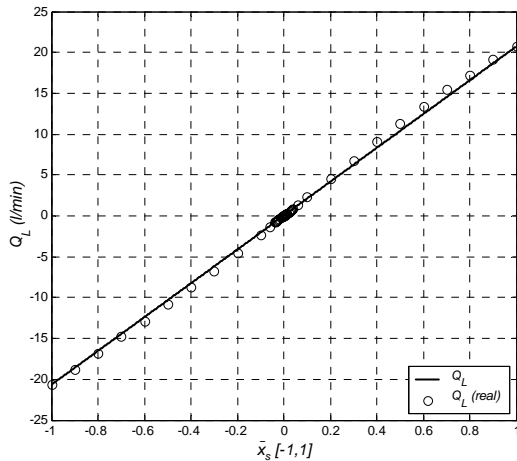


Figura 5.31 Caudal na carga para  $\bar{x}_s \hat{=} [-100\%, 100\%]$

### Modelo estático 5 (estrutura fixa)

Para as pseudo-seções de estrutura fixa utilizaram-se funções hiperbólicas [Ferreira02]. A principal razão para usar funções hiperbólicas é a possibilidade de caracterizar duas assíntotas bem definidas. Para uma válvula simétrica estas funções são:

$$\begin{aligned}
 A_n(\bar{x}_s) &= k_1 \bar{x}_s + k_2 + \sqrt{k_3 \bar{x}_s^2 + k_4 \bar{x}_s + k_5} \\
 A_p(\bar{x}_s) &= -k_1 \bar{x}_s + k_2 + \sqrt{k_3 \bar{x}_s^2 - k_4 \bar{x}_s + k_5}
 \end{aligned} \tag{5.81}$$

com  $k_3 \bar{x}_s^2 + k_4 \bar{x}_s + k_5 \geq 0$  e  $k_3 \bar{x}_s^2 - k_4 \bar{x}_s + k_5 \geq 0$  para  $\bar{x}_s \hat{=} [-1, 1]$  e com  $k_i \hat{=} \text{constante}$ .

As equações seguintes, relacionando as características estáticas da válvula e os parâmetros  $k_i$  de  $A_n(\bar{x}_s)$  e  $A_p(\bar{x}_s)$ , podem ser usadas para o cálculo dos parâmetros. O caudal nominal ( $Q_n$ ) e a pressão nominal ( $P_n$ ) podem usar-se para ajustar o caudal em  $\bar{x}_s = 1$ :

$$\frac{Q_n}{\sqrt{P_n}} = A_p(\bar{x}_s) \Big|_{\bar{x}_s=1} = -k_1 + k_2 + \sqrt{k_3 - k_4 + k_5} \tag{5.82}$$

Assumindo  $P_L = 0$ , pode ser obtida uma expressão para  $Q_L$ , válida para qualquer  $\bar{x}_s \hat{=} [-1, 1]$ , pelo uso simultâneo de (5.56) e (5.81):

$$Q_L(\bar{x}_s) = \sqrt{\frac{P_s}{2}} \left( -2k_1 \bar{x}_s + \sqrt{k_3 \bar{x}_s^2 - k_4 \bar{x}_s + k_5} - \sqrt{k_3 \bar{x}_s^2 + k_4 \bar{x}_s + k_5} \right) \tag{5.83}$$

Então, o ganho de caudal na posição central é:

$$\left. \frac{Q_L}{\bar{x}_s} \right|_{\bar{x}_s=0} = -\frac{2k_1 \sqrt{k_5} + k_4}{\sqrt{k_5}} \frac{\sqrt{P_s}}{\sqrt{2}} = \bar{K}_{q0} \tag{5.84}$$

Usando (5.58) e (5.81), o ganho de pressão relativa perto da posição intermédia da gaveta é dado por:

$$\left. \frac{q_{P_L}}{q_{\bar{x}_s}} \right|_{\bar{x}_s=0} = - \frac{2 \times k_1 \sqrt{k_5} + k_4}{k_2 \sqrt{k_5} + k_5} = \bar{K}_{p0} \quad (5.85)$$

O uso simultâneo de (5.62) e (5.81) leva a:

$$q_{lk0} = \sqrt{2 \times P_s} \times (k_2 + \sqrt{k_5}) \quad (5.86)$$

que pode ser usada para ajustar o caudal de fugas na posição central da gaveta.

Usando (5.63) podem ser obtidas novas expressões para o caudal de fugas e para a derivada do caudal de fugas, numa posição da gaveta afastada de posição central, onde a condição  $\bar{P}_L = 1$  pode ser assumida:

$$q_{lk}(\bar{x}_s)|_{\bar{P}_L=1} = 2 \times \sqrt{P_s} (k_1 \times \bar{x}_s + k_2 + \sqrt{k_3 \times \bar{x}_s^2 + \bar{x}_s \times k_4 + k_5}) \quad (5.87)$$

$$\left. \frac{q_{lk}}{q_{\bar{x}_s}} \right|_{\bar{P}_L=1} = 2 \times \sqrt{P_s} \frac{\partial}{\partial \bar{x}_s} k_1 + \frac{2k_3 \times \bar{x}_s + k_4}{2\sqrt{k_3 \times \bar{x}_s^2 + \bar{x}_s \times k_4 + k_5}} \frac{\partial}{\partial \bar{x}_s} \quad (5.88)$$

Se as curvas do caudal de fugas forem conhecidas, pode ser usada uma medida numa posição  $\bar{x}_s^* > 0$ ; se tal não for possível, o caudal de fugas e a sua derivada podem ser assumidos muito pequenos ou nulos em  $\bar{x}_s = 1$ .

Assumindo que a pressão relativa deve ser sempre reproduzida, os parâmetros para os modelos podem ser calculados através de (5.82), (5.85), (5.87), (5.88) e (5.84) ou (5.86) para a réplica do ganho de caudal ou do caudal de fugas, na posição central, respectivamente.

### Parametrização e simulações:

Os parâmetros do modelo, para ajuste do ganho de pressão relativa e do caudal de fugas, são calculados através de um sistema de equações não lineares definido pelas equações (5.82), (5.85),

(5.86), (5.87) e (5.88) e assumindo  $q_{lk}|_{\bar{x}_s=1} = 0$  e  $\left. \frac{q_{lk}}{q_{\bar{x}_s}} \right|_{\bar{x}_s=1} = 0$ . O conjunto de parâmetros para o

modelo é então:

$$k_1 = - 2.159; k_2 = - 1.083 \times 10^{-2}; k_3 = 4.675; k_4 = 1.535 \times 10^{-2}; k_5 = 1.582 \times 10^{-2};$$

As figuras 5.32, 5.33 e 5.34 mostram os resultados da simulação com os parâmetros anteriores.

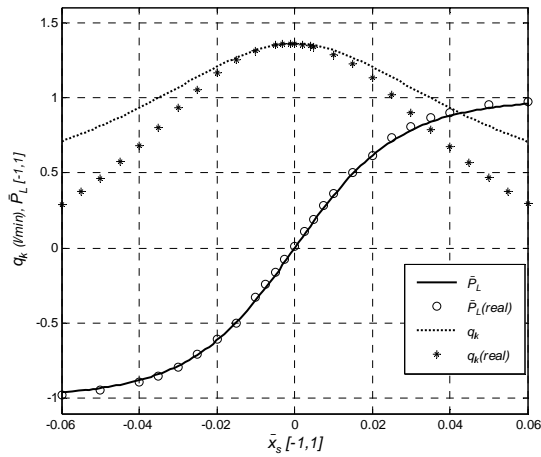


Figura 5.32 Caudal de fugas e ganho de pressão perto da posição intermédia:  $\bar{x}_s \hat{I} [-6\%, 6\%]$

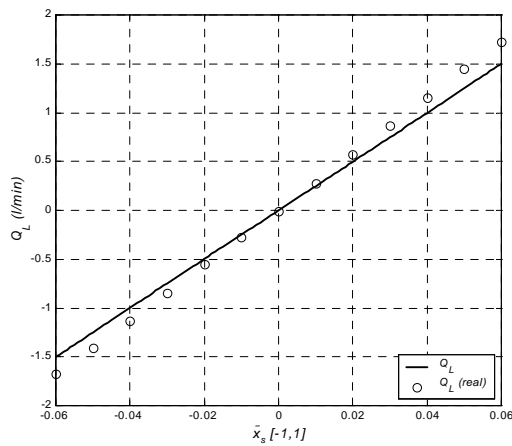


Figura 5.33 Caudal na carga perto da posição intermédia:  $\bar{x}_s \hat{I} [-6\%, 6\%]$

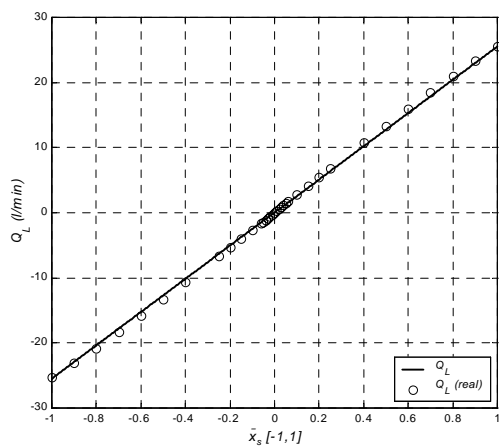


Figura 5.34 Caudal na carga para  $\bar{x}_s \hat{I} [-100\%, 100\%]$

As figuras 5.35, 5.36 e 5.37 apresentam os resultados da simulação quando é usada a pseudo-seccção com estrutura fixa, e seleccionando o ajuste do ganho de pressão e do ganho de caudal próximo da origem. Neste caso são assumidos  $q_{lk}|_{\bar{x}_s=1} = 0$  e  $\frac{q_{lk}}{\bar{x}_s}|_{\bar{x}_s=1} = 0$ . Usando (5.82), (5.84), (5.85), (5.87) e (5.88), obtêm-se os seguintes parâmetros para o modelo:

$$k_1 = -2.142, k_2 = 5.818 \times 10^{-3}, k_3 = 4.604, k_4 = -5.554 \times 10^{-2}, k_5 = 1.534 \times 10^{-2}.$$

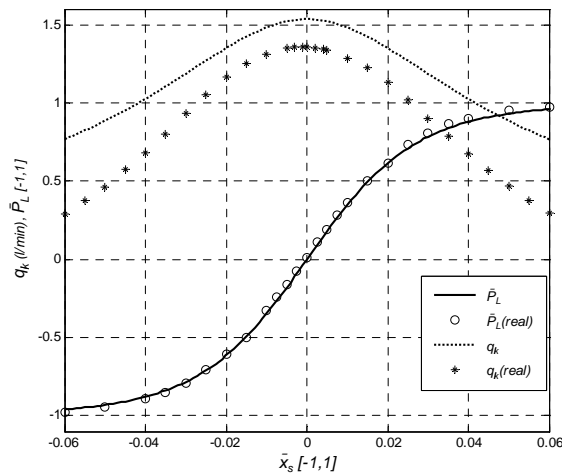


Figura 5.35 Caudal de fugas e ganho de pressão próximo da posição intermédia:  $\bar{x}_s \hat{I}$  [- 6%, 6% ]

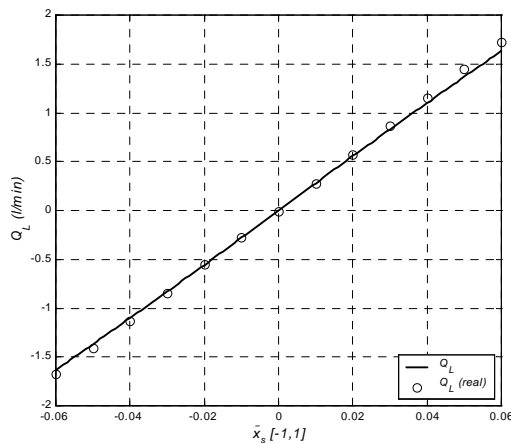


Figura 5.36 Caudal na carga perto da posição intermédia:  $\bar{x}_s \hat{I}$  [- 6%, 6% ]

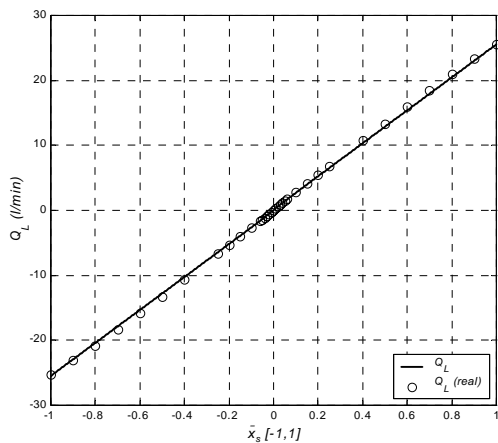


Figura 5.37 Caudal na carga para  $\bar{x}_s \hat{I} \in [-100\%, 100\%]$

### Comparação entre o modelo estático 4 e modelo estático 5

Ambas as aproximações apresentam resultados similares. As principais diferenças são a estrutura do modelo (variável ou fixa) e o esforço de computação necessário durante a simulação. O cálculo dos parâmetros  $k_i$  no modelo de estrutura fixa envolve a solução de um sistema de cinco equações não lineares. Em vez disso, somente é necessário resolver um sistema de três equações não lineares para o cálculo dos parâmetros do modelo de estrutura variável, sendo os outros parâmetros calculados através de equações algébricas. Durante a simulação, o modelo de estrutura fixa requer um tempo de cálculo constante e ligeiramente menor do que o requerido para o modelo de estrutura variável, como é desejado para as aplicações tempo real.

### Modelo estático 5 válido para pequenos sinais

Se o modelo for somente usado para pequenos valores (por exemplo  $\pm 10\%$  da máxima excursão da gaveta), podem ser ajustadas as curvas do caudal de fugas, do caudal na carga e do ganho relativo de pressão. Muito embora o ganho de caudal na origem não possa ser ajustado, este pode ser “quase” ajustado através de uma medição do caudal na carga próximo da origem, por exemplo 6%. Os resultados desta aproximação podem ver-se nas figuras 5.38 e 5.39.

Usando (5.82), (5.83), (5.85), (5.86), (5.87), e assumindo  $q_{lk}(0.06) = 0.293 \text{ l/min}$  e  $Q_L(0.06) = 1.72 \text{ l/min}$ , obtêm-se os seguintes valores para os parâmetros  $k_i$ :

$$k_1 = -2.788, k_2 = 6.107 \cdot 10^{-2}, k_3 = 2.506, k_4 = 7.438 \cdot 10^{-2}, k_5 = 2.902 \cdot 10^{-3}.$$



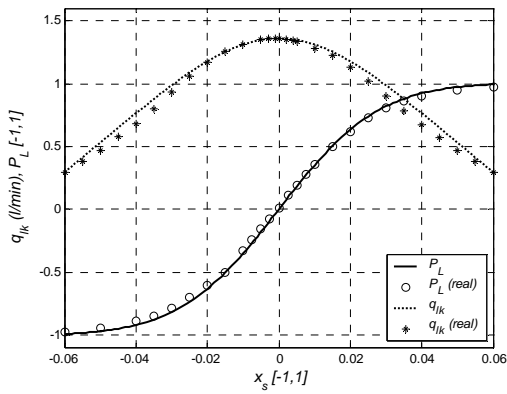


Figura 5.38 Caudal de fugas e ganho de pressão próximo da posição intermédia:  $\bar{x}_s \hat{I}$  [- 6%, 6% ]

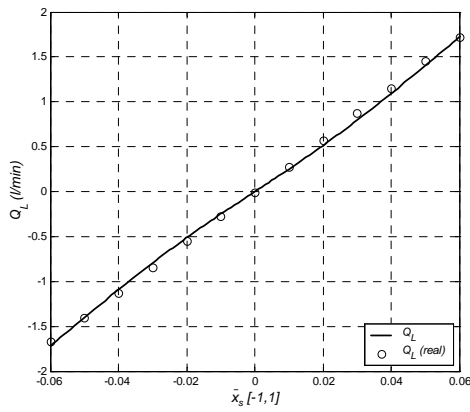


Figura 5.39 Caudal na carga perto da posição intermédia:  $\bar{x}_s \hat{I}$  [- 6%, 6% ]

A Figura 5.40 mostra os valores para  $A_n(\bar{x}_s)$  e  $A_p(\bar{x}_s)$  obtidos usando (5.60) e (5.61), com dados experimentais da pressão relativa na carga e do caudal de fugas, para  $-0.6 < \bar{P}_L < 0.6$  (este intervalo resulta das incertezas nas medidas de pressão e caudal).

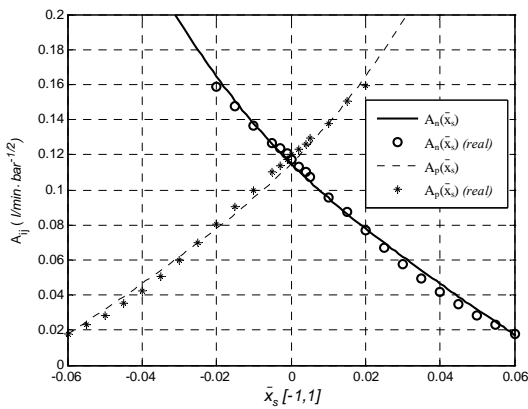


Figura 5.40 As pseudo-seções versus posição da gaveta

As pseudo-seccões foram calculadas pelo modelo de estrutura fixa, ajustando o caudal de fugas através de (5.87) e de uma medida do caudal de fugas em  $\bar{x}_s = 0.06$ . Pode ser observada uma aproximação bastante coincidente entre as pseudo-seccões medidas e as do modelo, Figura 5.40.

### Modelo estático 6 (estrutura variável)

Este modelo, de estrutura variável, utiliza a função hiperbólica e a função exponencial para a modelação das pseudo-seccões. Partindo dos parâmetros  $k_i$ , acima apresentados para as pseudo-seccões, é possível usar o modelo em toda a gama de funcionamento da gaveta da válvula, se se comutar a pseudo-seccão em, por exemplo,  $\bar{x}_s = 0.06$ . Assim, pode-se utilizar a função exponencial garantindo a continuidade de  $A_p(\bar{x}_s)$  e  $A_n(\bar{x}_s)$  e das suas derivadas no ponto de transição:

$$A_n(\bar{x}_s) = \begin{cases} k_1 \bar{x}_s + k_2 + \sqrt{k_3 \bar{x}_s^2 + k_4 \bar{x}_s + k_5} & ; \bar{x}_s \leq x_c \\ b_1 \times e^{-b_2 \bar{x}_s} & ; \bar{x}_s > x_c \end{cases}$$

$$A_p(\bar{x}_s) = \begin{cases} -k_1 \bar{x}_s + k_2 + \sqrt{k_3 \bar{x}_s^2 - k_4 \bar{x}_s + k_5} & ; \bar{x}_s \geq x_c \\ b_1 \times e^{-b_2 \bar{x}_s} & ; \bar{x}_s < x_c \end{cases} \quad (5.89)$$

Então, se  $x_c = 0.06$ ,  $b_1 = 2.253$  e  $b_2 = 80.095$ , a continuidade das pseudo-seccões e das suas derivadas é assegurada. A Figura 5.41 mostra as pseudo-seccões para todo o curso da gaveta.

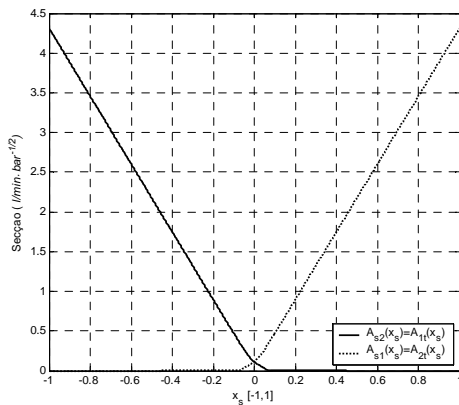


Figura 5.41 As pseudo-seccões versus posição da gaveta para  $\bar{x}_s \in [-1, 1]$

As figuras 5.42, 5.43 e 5.44 ilustram os resultados para o caudal de fugas, o ganho relativo de pressão e o ganho de caudal.

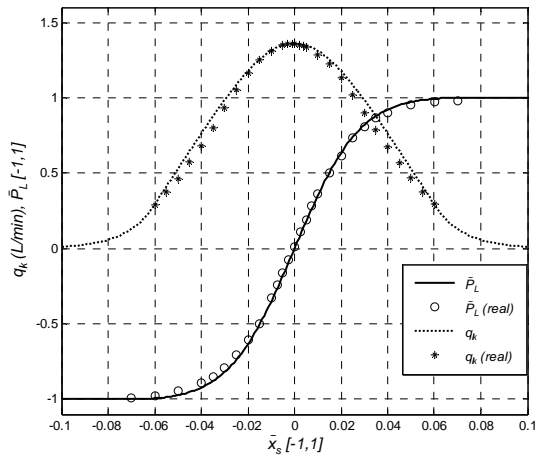


Figura 5.42 Caudal de fugas e ganho de pressão próximo da posição intermédia:  $\bar{x}_s \hat{I} [-10\%, 10\%]$

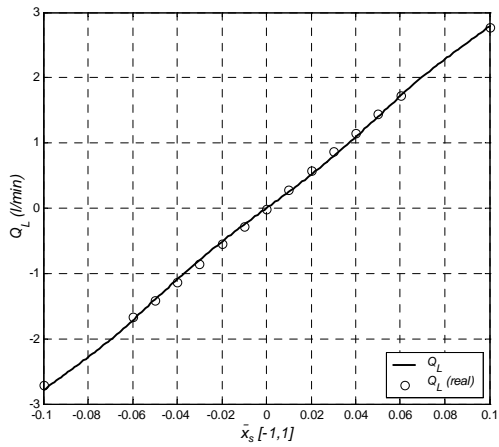


Figura 5.43 Caudal na carga perto da posição intermédia:  $\bar{x}_s \hat{I} [-10\%, 10\%]$

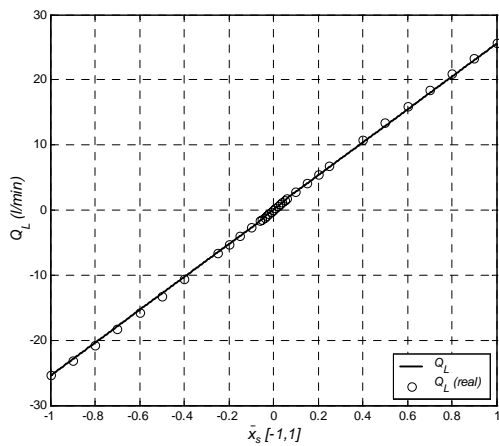


Figura 5.44 Caudal na carga:  $\bar{x}_s \hat{I} [-100\%, 100\%]$

## Parametrização e simulações (válvula modelo NG6 da Bosch®)

A seguir (figuras 5.45 a 5.54) apresentam-se algumas simulações dos modelos com pseudo-seções baseados nas funções hiperbólicas (modelo estático 5 e modelo estático 6).

Todas as simulações usam os seguintes valores (extraídos de curvas experimentais) para a válvula modelo NG6 da Bosch (Ref: O811404038):

$$Q_n = 20.5 \text{ l/min}, P_n = 35 \text{ bar}, P_s = 100 \text{ bar}, \bar{K}_{q0} = 22, \bar{K}_{p0} = 74, q_{lk0} = 0.358 \text{ l/min}$$

Para a reprodução do ganho de pressão relativo e do caudal de fugas máximo usa-se (5.82), (5.85),

(5.86), (5.87) e (5.88) e, assumindo  $q_{lk}|_{\bar{x}_s=1} = 0$  e  $\left. \frac{dq_{lk}}{d\bar{x}_s} \right|_{\bar{x}_s=1} = 0$ , os parâmetros são os seguintes:

$$k_1 = -1.773, k_2 = -2.357 \cdot 10^{-2}, k_3 = 3.142, k_4 = 8.172 \cdot 10^{-2}, k_5 = 2.390 \cdot 10^{-3}$$

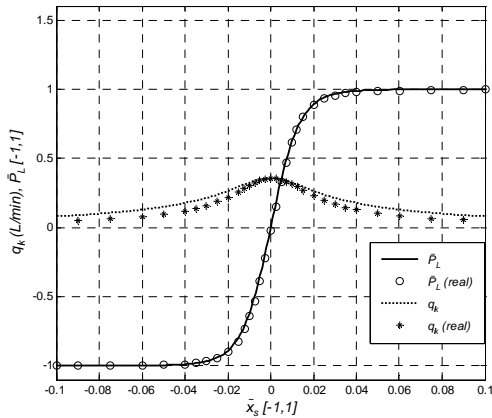


Figura 5.45 Caudal de fugas e ganho de pressão relativo (ajuste de  $q_{lk0}$  e  $\bar{K}_{p0}$ )

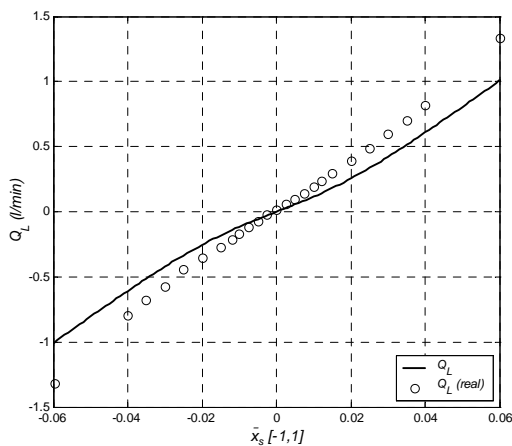


Figura 5.46 Caudal na carga (ajuste de  $q_{lk0}$  e  $\bar{K}_{p0}$ )

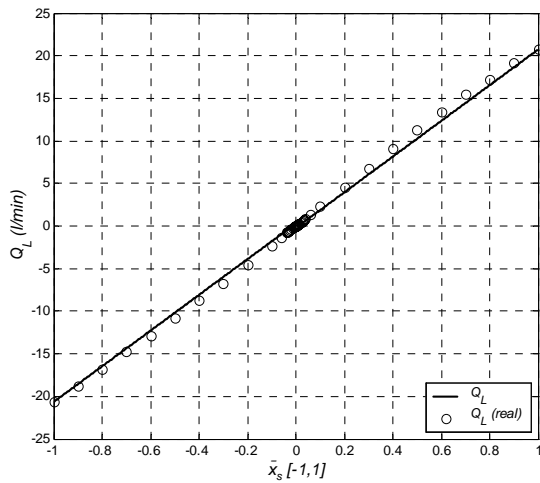


Figura 5.47 Caudal na carga (ajuste de  $q_{lk0}$  e  $\bar{K}_{p0}$ )

Usando (5.82), (5.85), (5.84), (5.87) e (5.88), para ajuste do ganho de caudal, e assumindo

$q_{lk} |_{\bar{x}_s=1} = 0$  e  $\left. \frac{q_{lk}}{\bar{x}_s} \right|_{\bar{x}_s=1} = 0$ , são obtidos os seguinte parâmetros:

$$k_1 = -1.755, k_2 = -6.898 \cdot 10^{-2}, k_3 = 3.082, k_4 = 1.952 \cdot 10^{-2}, k_5 = 2.395 \cdot 10^{-3}.$$

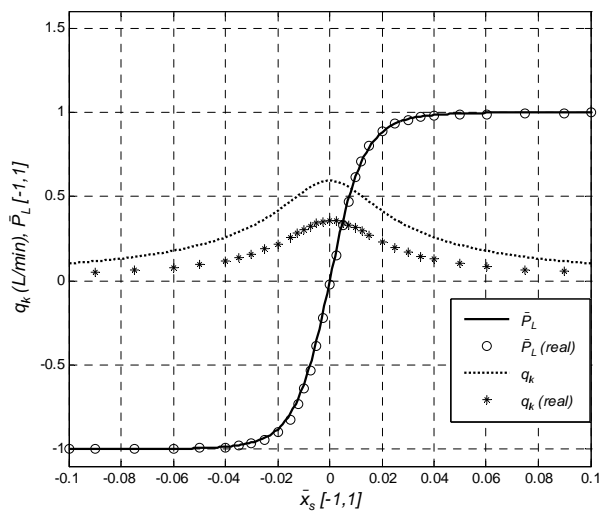


Figura 5.48 Caudal de fugas e ganho de pressão relativo (ajuste de  $q_{lk0}$  e  $\bar{K}_{q0}$ )

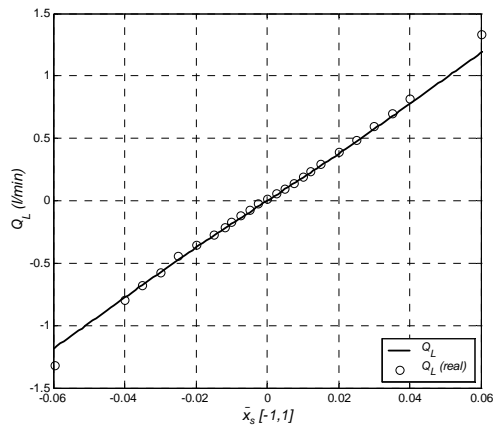


Figura 5.49 Caudal na carga (ajuste de  $q_{lk0}$  e  $\bar{K}_{q0}$ )

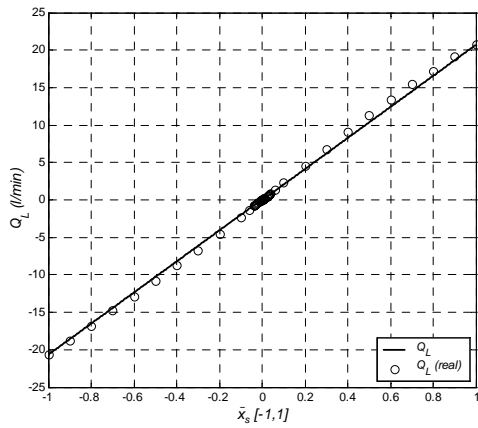


Figura 5.50 Caudal na carga (ajuste de  $q_{lk0}$  e  $\bar{K}_{q0}$ )

Com o modelo estático 6, usando para as pseudo-seções as expressões em (5.89) com  $x_c = 0.035$ ,  $q_{lk}(0.035) = 0.145$  l/min e  $Q_L(0.035) = 0.7$  l/min, resultam os seguintes valores para os parâmetros:

$$k_1 = -1.955, k_2 = -1.193 \cdot 10^{-2}, k_3 = 2.375, k_4 = 2.741 \cdot 10^{-2}, k_5 = 1.830 \cdot 10^{-4}, \text{ e}$$

$$b_1 = -1.555 \cdot 10^{-2}, b_2 = 85.937$$

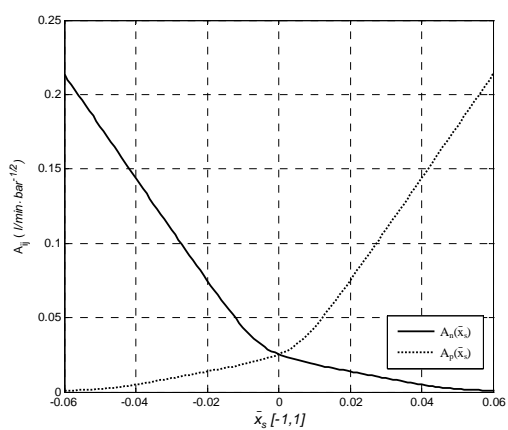


Figura 5.51 As pseudo-seções versus posição da gaveta para  $\bar{x}_s \hat{I} [-6\%, 6\%]$

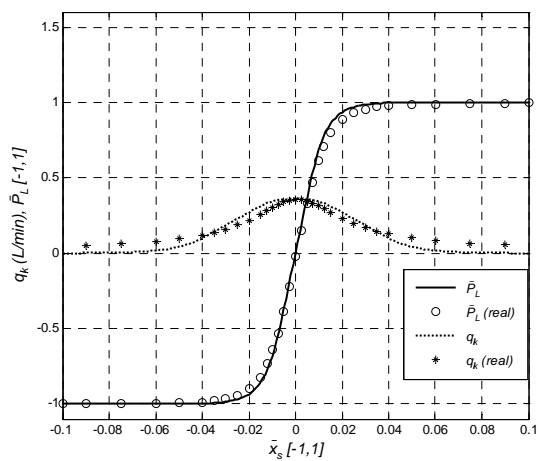


Figura 5.52 Caudal de fugas e ganho de pressão relativo (modelo estático 6)

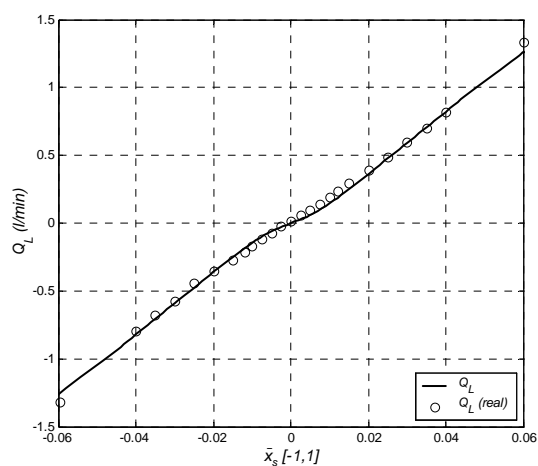


Figura 5.53 Caudal na carga (modelo estático 6)

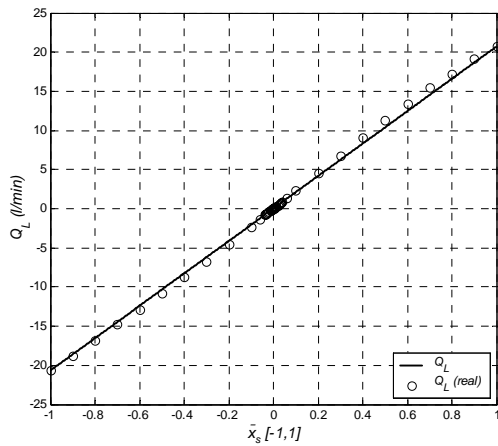


Figura 5.54 Caudal na carga (modelo estático 6)

### Modelo estático 5a (estrutura fixa)

Uma característica de algumas válvulas é serem simétricas e centradas, mas com orifícios desiguais. No caso da válvula, modelo KBSDG4V-3 da Vickers, os orifícios da câmara da válvula para os terminais ligados ao tanque são simétricos e emparelhados, o mesmo acontecendo com os orifícios do terminal fonte para as câmaras da válvula. As pseudo-seções podem então ser feitas simétricas duas a duas, isto é  $A_{s1}(\bar{x}_s) = A_{s2}(-\bar{x}_s)$  e  $A_{1t}(\bar{x}_s) = A_{2t}(-\bar{x}_s)$ .

Usando as pseudo-seções com estrutura fixa (funções hiperbólicas), têm-se as seguintes funções para as pseudo-seções:

$$\begin{aligned}
 A_{s2}(\bar{x}_s) &= k_1 \bar{x}_s + k_2 + \sqrt{k_3 \bar{x}_s^2 + k_4 \bar{x}_s + k_5} \\
 A_{s1}(\bar{x}_s) &= -k_1 \bar{x}_s + k_2 + \sqrt{k_3 \bar{x}_s^2 - k_4 \bar{x}_s + k_5} \\
 A_{1t}(\bar{x}_s) &= k_{1t} \bar{x}_s + k_{2t} + \sqrt{k_{3t} \bar{x}_s^2 + k_{4t} \bar{x}_s + k_{5t}} \\
 A_{2t}(\bar{x}_s) &= -k_{1t} \bar{x}_s + k_{2t} + \sqrt{k_{3t} \bar{x}_s^2 - k_{4t} \bar{x}_s + k_{5t}}
 \end{aligned} \tag{5.90}$$

com  $k_3 \bar{x}_s^2 \pm k_4 \bar{x}_s + k_5 \geq 0$  e  $k_{3t} \bar{x}_s^2 \pm k_{4t} \bar{x}_s + k_{5t} \geq 0$  para  $\bar{x}_s \in [-1, 1]$  e com  $k_i, k_{it} \geq 0$ .

### Parametrização

Os parâmetros  $k_i$  de  $A_{s1}(\bar{x}_s)$ ,  $A_{s2}(\bar{x}_s)$ ,  $A_{1t}(\bar{x}_s)$ ,  $A_{2t}(\bar{x}_s)$ , também podem ser calculados através dos manuais do fabricante ou por meio de medidas experimentais, de modo a reproduzir o ganho de pressão relativa, o ganho de caudal ou o caudal de fugas na origem. Na parametrização do modelo usam-se as seguintes características estáticas medidas experimentalmente:

$$Q_n = 25.5 \text{ l/min}, P_n = 35 \text{ bar}, P_s = 70 \text{ bar}, \bar{K}_{q0} = 28 \text{ l/min}, \bar{K}_{p0} = 36.5, q_{lk0} = 1.36 \text{ l/min}$$



Em algumas válvulas as pressões nas câmaras podem não se interceptar em  $P_s/2$ , como se pode ver na Figura 5.55. A válvula em estudo tem o ponto de intercepção em 44 bar para  $P_s=70$  bar, tendo então uma diferença de  $DP_m = 9$  bar relativa a  $P_s/2$ . Usando as equações (5.51) e (5.6) e considerando a montagem representada na Figura 5.6 (montagem de medição do ganho de pressão), isto é  $Q_1=Q_2=0$ , pode ser estabelecida a seguinte relação para a diferença de pressão entre  $P_1$  e  $P_2$  na posição intermédia da gaveta:

$$DP_m = \frac{P_s A_{s2} (0)^2 - A_{2t} (0)^2}{2 A_{s2} (0)^2 + A_{2t} (0)^2} \quad (5.91)$$

Usando novamente as equações (5.51) e (5.6) e  $Q_1=Q_2 = 0$ , a pressão relativa na carga pode ser definida por:

$$\overline{P}_L(x_s) = \frac{A_{s1} (\overline{x}_s)^2}{A_{s1} (\overline{x}_s)^2 + A_{1t} (\overline{x}_s)^2} - \frac{A_{s2} (\overline{x}_s)^2}{A_{s2} (\overline{x}_s)^2 + A_{2t} (\overline{x}_s)^2} \quad (5.92)$$

onde  $P_L = P_1 - P_2$  and  $\overline{P}_L = \frac{P_L}{P_s}$

O ganho de pressão é então:

$$\overline{K}_{p0} = \left. \frac{\overline{P}_L(\overline{x}_s)}{\overline{x}_s} \right|_{\overline{x}_s=0} \quad (5.93)$$

Na medição do ganho de caudal (montagem ilustrada na Figura 5.5), que corresponde a ligar os terminais 1 e 2 com uma resistência hidráulica nula, o caudal na carga,  $Q_L$ , pode ser expresso por  $Q_1$  ou por  $Q_2$ . Assim, usando (5.51) e (5.6) e a diferença de pressão das câmaras para o ponto intermédio,  $DP_m$ , o ganho de caudal é dado pela seguinte expressão (usando  $Q_L=Q_1$ ):

$$Q_L(\overline{x}_s) = A_{s1}(\overline{x}_s) \sqrt{\frac{P_s}{2} - DP_m} - A_{1t}(\overline{x}_s) \sqrt{\frac{P_s}{2} + DP_m} \quad (5.94)$$

donde o ganho de caudal pode ser definido por:

$$\overline{K}_{q0} = \left. \frac{Q_L(\overline{x}_s)}{\overline{x}_s} \right|_{\overline{x}_s=0} = \left. \frac{A_{p}(\overline{x}_s)}{\overline{x}_s} \right|_{\overline{x}_s=0} \sqrt{\frac{P_s}{2} - DP_m} - \left. \frac{A_{nt}(\overline{x}_s)}{\overline{x}_s} \right|_{\overline{x}_s=0} \sqrt{\frac{P_s}{2} + DP_m} \quad (5.95)$$

Usando  $Q_L=Q_2$ , o ganho de caudal pode ser expresso por:

$$\overline{K}_{q0} = \left. \frac{A_{2t}(\overline{x}_s)}{\overline{x}_s} \right|_{\overline{x}_s=0} \sqrt{\frac{P_s}{2} + DP_m} - \left. \frac{A_{s2}(\overline{x}_s)}{\overline{x}_s} \right|_{\overline{x}_s=0} \sqrt{\frac{P_s}{2} - DP_m} \quad (5.96)$$

O caudal na carga fora da região intermédia pode ser ajustado através do caudal nominal ( $Q_n$ ) e da pressão nominal ( $P_n$ ) que, ou são disponibilizados pelo manual do fabricante, ou podem ser medidos para uma válvula específica. Isto é verdade porque  $A_{s2}(\bar{x}_s) \gg A_{1t}(\bar{x}_s) \gg 0$  para  $\bar{x}_s = 1$ .

$$\frac{Q_n}{\sqrt{P_n}} = A_{s1}(\bar{x}_s) \Big|_{\bar{x}_s=1} \quad (5.97)$$

$$\frac{Q_n}{\sqrt{P_n}} = A_{2t}(\bar{x}_s) \Big|_{\bar{x}_s=1} \quad (5.98)$$

O caudal de fugas pode ser expresso como uma função da pressão relativa nas câmaras da válvula,  $\bar{P}_1 = P_1/P_s$  e  $\bar{P}_2 = P_2/P_s$ , usando as equações (5.51) e (5.6) com  $Q_1=Q_2=0$ :

$$q_{lk}(\bar{x}_s) = q_{s1} + q_{s2} = A_{s1}(\bar{x}_s) \sqrt{P_s(1-\bar{P}_1)} + A_{s2}(\bar{x}_s) \sqrt{P_s(1-\bar{P}_2)} \quad (5.99)$$

$$q_{lk}(\bar{x}_s) = q_{1t} + q_{2t} = A_{1t}(\bar{x}_s) \sqrt{P_s \bar{P}_1} + A_{2t}(\bar{x}_s) \sqrt{P_s \bar{P}_2} \quad (5.100)$$

Assumindo as condições de  $\bar{P}_1 \gg 1$ ,  $\bar{P}_2 \gg 0$  e usando (5.99) e (5.100), podem ser estabelecidas novas relações para o caudal de fugas e a sua derivada numa determinada posição da gaveta. Se a curva experimental do caudal de fugas estiver disponível, pode ser usada uma medida numa determinada posição ( $\bar{x}_s > 0$ ), caso contrário pode ser atribuído um valor pequeno, ou mesmo nulo, ao caudal de fugas em  $\bar{x}_s = 1$ .

$$q_{lk} \Big|_{\substack{\bar{P}_1=1 \\ \bar{P}_2=0}} = \sqrt{P_s} A_{s2}(\bar{x}_s) \quad (5.101)$$

$$q_{lk} \Big|_{\substack{\bar{P}_1=1 \\ \bar{P}_2=0}} = \sqrt{P_s} A_{1t}(\bar{x}_s) \quad (5.102)$$

$$\frac{\partial q_{lk}}{\partial \bar{x}_s} \Big|_{\substack{\bar{P}_1=1 \\ \bar{P}_2=0}} = \sqrt{P_s} \frac{dA_{s2}(\bar{x}_s)}{d\bar{x}_s} \Big|_{\substack{\bar{P}_1=1 \\ \bar{P}_2=0}} \quad (5.103)$$

$$\frac{\partial q_{lk}}{\partial \bar{x}_s} \Big|_{\substack{\bar{P}_1=1 \\ \bar{P}_2=0}} = \sqrt{P_s} \frac{dA_{1t}(\bar{x}_s)}{d\bar{x}_s} \Big|_{\substack{\bar{P}_1=1 \\ \bar{P}_2=0}} \quad (5.104)$$

Usando (5.90), podem ser definidas dez equações para calcular os parâmetros  $k_i$  e  $k_{it}$  das equações que definem as pseudo-secções. Então, usando (5.91), (5.93), (5.95), (5.96), (5.97), (5.98), (5.101), (5.102), (5.103) e (5.104), com  $DP_m = 9 \text{ bar}$ , e considerando que os caudais de fugas e as suas

derivadas são nulas para  $\bar{x}_s = 1$  ( $\bar{P}_1 = 1, \bar{P}_2 = 0$ ), os parâmetros para as pseudo-seções em (5.90) são os seguintes:

$$k_1 = -2.129, k_2 = 1.860 \cdot 10^{-2}, k_3 = 4.550, k_4 = -1.094 \cdot 10^{-1}, k_5 = 1.548 \cdot 10^{-2},$$

$$k_{1t} = -2.146, k_{2t} = 3.393 \cdot 10^{-3}, k_{3t} = 4.619, k_{4t} = -3.724 \cdot 10^{-2}, k_{5t} = 1.548 \cdot 10^{-2}$$

Neste caso são ajustados os ganhos relativos de pressão e o ganho de caudal. Para ajustar o caudal de fugas na origem devem ser usadas as equações (5.99) e (5.100) em vez das equações de ajuste do ganho de caudal (5.95) e (5.96). As figuras 5.55 a 5.59 ilustram as curvas de pressão relativas, as diferenças nas pseudo-seções e os ganhos de caudal, para uma simulação do modelo com os parâmetros acima apresentados.

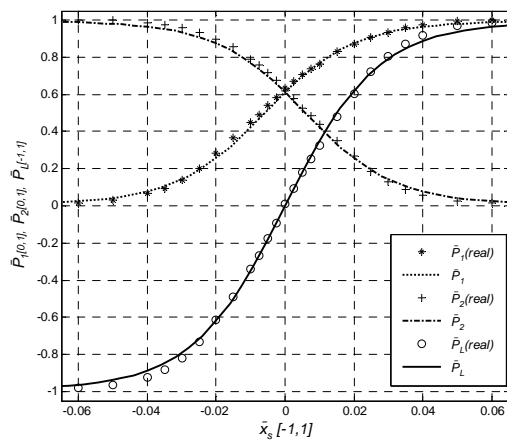


Figura 5.55 Ganho de pressão relativa, pressão relativa ( $\bar{P}_1, \bar{P}_2$ ) nos terminais 1 e 2

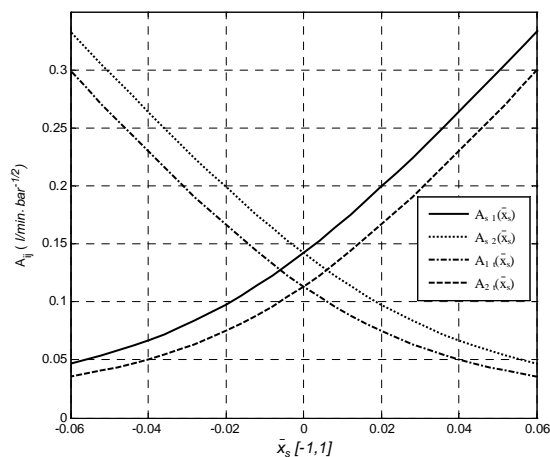


Figura 5.56 Pseudo seções para  $\bar{x}_s \hat{I}$  [- 6%, 6% ]

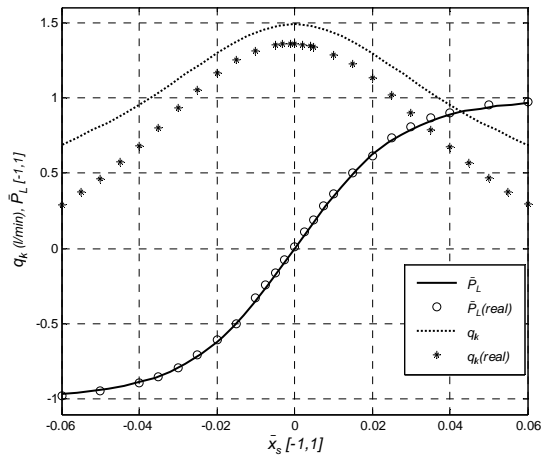


Figura 5.57 Ganho de pressão relativa e caudal de fugas (acerto de  $\bar{K}_{p0}$  e  $\bar{K}_{q0}$ )

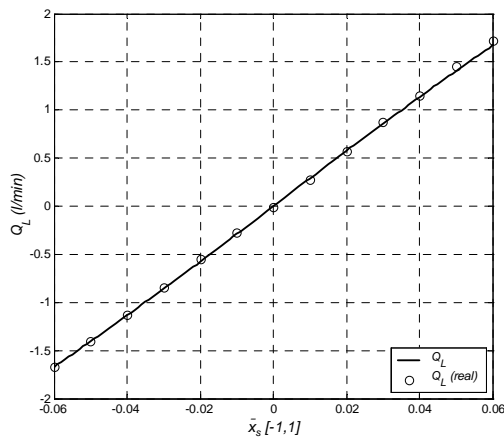


Figura 5.58 Caudal na carga para  $\bar{x}_s \hat{I} [- 6\%, 6\% ]$

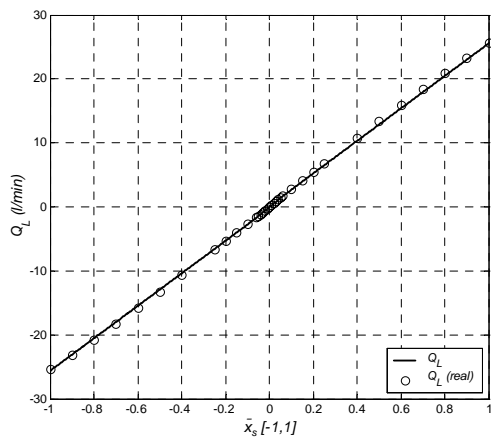


Figura 5.59 Caudal na carga para  $\bar{x}_s \hat{I} [- 100\%, 100\% ]$

De entre todos os modelos estáticos apresentados nesta secção, os que apresentam menor desempenho são aqueles em que os caudais apresentam uma componente laminar e outra turbulenta (modelos estáticos 1 e 2), não só porque não reproduzem adequadamente as características estáticas da válvula, mas também porque as derivadas de algumas das funções, que descrevem essas características, não são contínuas. Os modelos baseados em pseudo-secções, somente com escoamentos turbulentos, são aqueles que apresentam melhores aproximações para as características estáticas. À parte o modelo estático 3, todos os outros modelos, baseados em pseudo-secções com escoamentos turbulentos, permitem ajustar simultaneamente dois de três parâmetros estáticos (ganho de pressão, ganho de caudal ou caudal de fugas na posição intermédia). Os modelos estáticos 4 e 5, mesmo sendo um de estrutura variável e o outro de estrutura fixa, apresentam desempenhos equivalentes. O modelo estático 5 contempla algumas variações e/ou extensões: a mais simples, que ajusta o caudal de fugas máximo e um outro ponto da característica do caudal de fugas, permite obter desempenhos muito bons para os três parâmetros, muito embora só seja válido para pequenos sinais. Uma outra extensão, representada no modelo estático 6, permite, à custa da introdução duma estrutura variável nas pseudo-secções, usar as características da extensão anterior para todo o curso da válvula; a última extensão apresentada, modelo estático 5a, estende o modelo estático 5 para orifícios desiguais, muito embora necessite de dez parâmetros. O modelo estático 5 pode ainda ser estendido com uma pseudo-secção diferente para cada orifício sem afectar o tempo de cálculo na simulação. Contudo, a parametrização de um tal modelo, necessitaria de um cálculo de 20 parâmetros.

## 5.2 Modelos dinâmicos

Os modelos dinâmicos apresentados nesta secção são orientados para aplicações de controlo. Como a histerese neste tipo de válvulas é normalmente menor que 0.5% (valor fornecido pelos fabricantes), e outros distúrbios, como as forças de atrito na gaveta e as forças devidas ao escoamento, são minimizados pelo controlo em malha fechada da posição da gaveta, o comportamento mais importante a ser modelado é a resposta em frequência do movimento da gaveta em relação a um determinado sinal de entrada. Na Figura 5.60 é apresentado um diagrama de Bode típico, fornecido pelo fabricante, para a válvula proporcional modelo KBSDD4V-3 da Vickers. Normalmente são fornecidos diagramas para diferentes amplitudes do movimento em torno da posição intermédia.

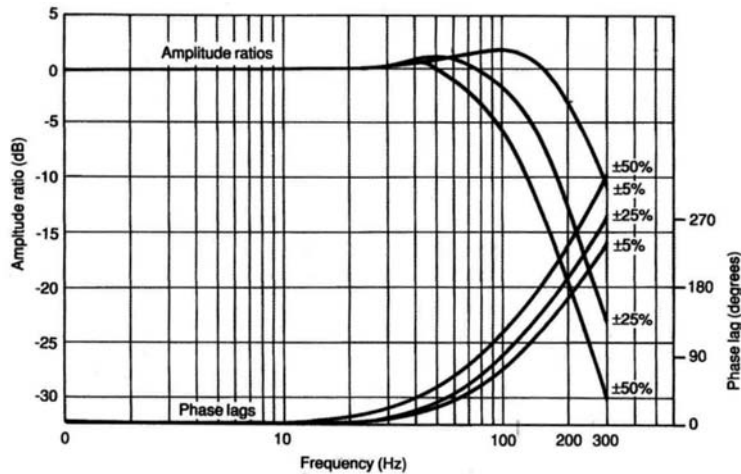


Figura 5.60 Resposta na frequência típica para  $\pm 5\%$ ,  $\pm 25\%$  e  $\pm 50\%$  do curso máximo da gaveta (cortesia da Eaton Corporation)

Pretende-se averiguar o desempenho de um modelo de 2ª ordem na modelação da dinâmica dominante do órgão móvel da válvula. Este procedimento foi seguido em vários trabalhos, nomeadamente em [Virtalo97] e [Quintas99]. Quintas, em [Quintas99], usa um modelo de 2ª ordem com coeficiente de amortecimento,  $\zeta$ , frequência angular natural,  $\omega_n$ , e com um atraso,  $\Delta t$ , para compensação da fase, Figura 5.61.

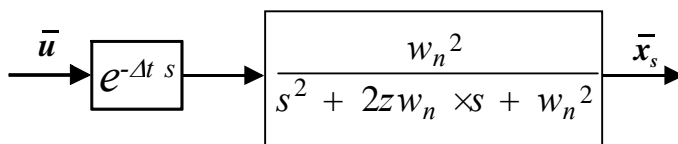


Figura 5.61 Modelo de 2ª ordem com atraso para compensação da fase [Quintas99]

O ajuste mais simples dos parâmetros do modelo de 2ª ordem é preconizado por Virtalo [Virtalo97] e consiste no ajuste da curva de  $\pm 5\%$ . O método propõe a medição da frequência correspondente a um atraso de fase de  $-90^\circ$ , obtendo-se a frequência angular natural do sistema,  $\omega_n = 2\pi f_n$  e o coeficiente de amortecimento  $z = \frac{1}{2|G|_{\omega_n, 90}}$ .  $|G|_{\omega_n, 90}$  é o ganho lido na curva de  $\pm 5\%$  para a frequência cujo atraso de fase é  $-90^\circ$ .

Usando os valores medidos  $f_n = 115\text{Hz}$ ,  $|G|_{\omega_n, 90} = 1.382$ , obtém-se  $\omega_n = 723\text{rad} \times \text{s}^{-1}$ ;  $z = 0.364$ . Os resultados desta parametrização são ilustrados na Figura 5.62. Usando um método de optimização, por exemplo, o método *simplex* proposto em [Lagarias98], e ajustando a curva do ganho de amplitude, obtém-se os resultados ilustrados na Figura 5.63.

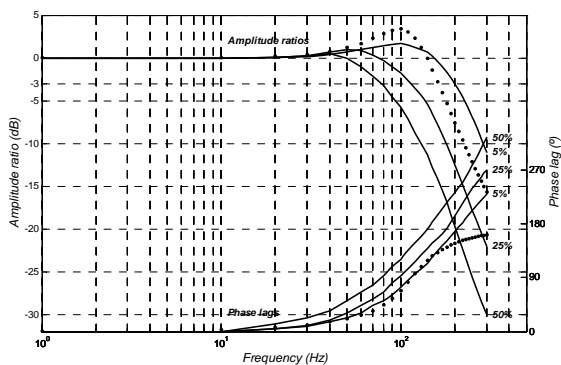


Figura 5.62 Diagrama de Bode para  $w_n = 723\text{rad} \times \text{s}^{-1}$  e  $z = 0.364$ , obtidos pelo método de [Virtalo98]

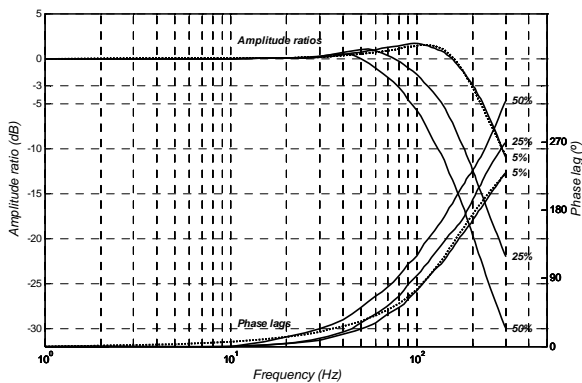


Figura 5.63 Diagrama de Bode para  $w_n = 954\text{rad} \times \text{s}^{-1}$ ,  $z = 0.480$  e  $Dt = 7.4 \times 10^{-4}\text{s}$  obtidos por otimização

O diagrama de Bode de amplitude sugere a introdução de não linearidades no modelo de 2ª ordem, de modo a “reduzir” a dinâmica do movimento para maiores excursões do sinal introduzido na carta de controlo,  $\bar{u}$ . A não linearidade mais simples, sugerida por Beater, em [Beater98a], consiste na limitação de velocidade da gaveta. Os modelos dinâmicos apresentados nas secções seguintes relatam diferentes não linearidades introduzidas no modelo de 2ª ordem do modelo da Figura 5.61. Para ilustração do método usado para obter os parâmetros do modelo, neste caso  $w_n, x, L_v, L_a$  e  $Dt$  utiliza-se o modelo da Figura 5.64. Este modelo contém duas não linearidades distintas: saturação de velocidade,  $L_v$ , e de aceleração,  $L_a$ . O atraso  $\Delta t$  é usado para ajuste da fase da resposta.

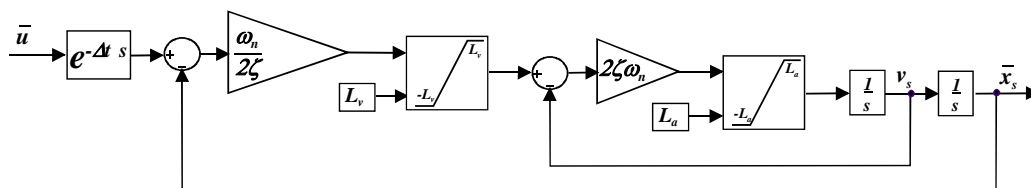


Figura 5.64 Modelo dinâmico do movimento da gaveta com limitação de velocidade e aceleração

### 5.2.1 Ajuste do ganho de amplitude

Os parâmetros do modelo (frequência natural,  $\omega_n$ , coeficiente de amortecimento,  $\xi$ , limite de velocidade,  $L_v$ , e limite de aceleração,  $L_a$ ) são estimados através do método *simplex* proposto em [Lagarias98] e implementado na função *fminsearch* da *Optimization Toolbox* [Coleman99] fornecida na plataforma Matlab. A função de custo,  $F$  (5.107), utiliza o método dos mínimos quadrados. No exemplo presente, o modelo foi simulado numa gama de frequências de 10 a 200 Hz em passos de 10 Hz. Os parâmetros foram otimizados usando as três curvas de ganho de amplitude do diagrama de Bode fornecido pelo fabricante (5%, 25% e 50% da amplitude máxima).

Foi aplicada uma onda sinusoidal de frequência (e amplitude) variável,  $\bar{u}(t)$ , à entrada do modelo dinâmico do movimento da gaveta:

$$\bar{u}(t) = A_{in} \times \sin(2p \times f_k \times t) \quad (5.105)$$

onde o inteiro  $k \in [1, 20]$ ,  $f_k = 10 \times k$  e  $t$  representa a variável tempo.

A posição de saída do modelo,  $\bar{x}_s(t)$ , foi então filtrada com um filtro digital passa banda, com uma banda passante estreita e fase nula, de modo a excluir os harmónicos e sub-harmónicos e, em seguida, foi usada para calcular o ganho,  $G_{s_k}$ , em dB:

$$G_s = 20 \times \log_{10} \left( \frac{A_{out}}{A_{in}} \right) \quad (5.106)$$

em que  $A_{out}$  é a amplitude da sinusóide fundamental da saída.

O ganho é depois comparado com o ganho fornecido pelo fabricante, à mesma frequência e amplitude ( $G_{r_k}$ ) calculando-se, de seguida, a função de custo,  $F$ . Como na maior parte do tempo a válvula actua em torno da posição intermédia, pode-se dar mais ênfase à curva de  $\pm 5\%$ , através da atribuição de pesos diferentes no erro quadrático para o cálculo do valor da função de custo. No caso ilustrado na expressão (5.107) atribuiu-se um peso 4 ao ajuste da curva de  $\pm 5\%$  e um peso unitário para as outras curvas:

$$F(\omega_n, \xi, L_v, L_a) = 4 \times \sum_{k=1}^{20} \left. \frac{(G_{s_k} - G_{r_k})^2}{2} \right|_{A_{in}=5\%} + \sum_{k=1}^{20} \left. \frac{(G_{s_k} - G_{r_k})^2}{2} \right|_{A_{in}=25\%} + \sum_{k=1}^{20} \left. \frac{(G_{s_k} - G_{r_k})^2}{2} \right|_{A_{in}=50\%} \quad (5.107)$$

Com base nos parâmetros anteriores e no novo valor para a função de custo, o algoritmo de optimização gera, então, novos parâmetros, sendo iniciada uma nova iteração, de (5.105) a (5.107).



As iterações terminam quando os parâmetros dos modelos convergem para um conjunto de valores que minimizam a função de custo.

## 5.2.2 Modelos e simulações

Nesta secção apresentam-se alguns modelos e os seus diagramas de Bode (amplitude), simulados para os parâmetros estimados pelo método apresentado na secção anterior. São apresentados modelos dinâmicos contendo não linearidades com diferentes complexidades, entre os quais modelos com saturação de força disponível, de potência disponível, de velocidade e aceleração. Foram também testados diversos limitadores, com diferentes características não lineares, baseadas em funções contínuas e com estrutura variável.

### Modelo dinâmico 1

O modelo mais simples, ilustrado Figura 5.65, consiste na saturação de velocidade através do parâmetro,  $L_v$ . O diagrama de Bode correspondente a este modelo é apresentado na Figura 5.66.

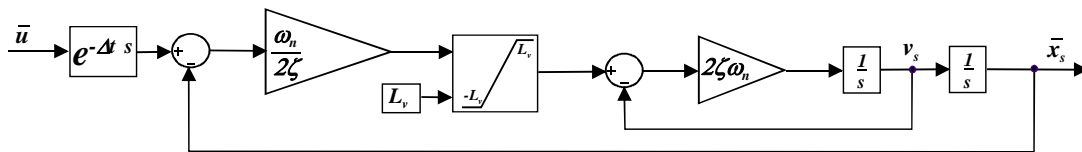


Figura 5.65 Modelo dinâmico com saturação de velocidade,  $L_v$

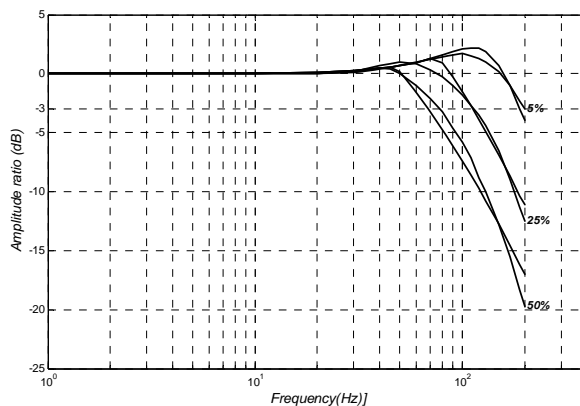


Figura 5.66 Diagrama de Bode (amplitude) para  $f_n = 141.4 \text{ s}^{-1}$ ;  $x = 0.431$ ;  $L_v = 135.7 \text{ s}^{-1}$ ; ( $F = 17.968$ )

Um modelo equivalente, com uma limitação de velocidade baseada na função arco-tangente, e avaliado pelo valor devolvido pela função de custo,  $F = 15.982$ , melhora ligeiramente o desempenho do modelo.

## Modelo dinâmico 2

Este modelo, Figura 5.67, implementa uma saturação abrupta de força através do parâmetro,  $F_{max}$ .

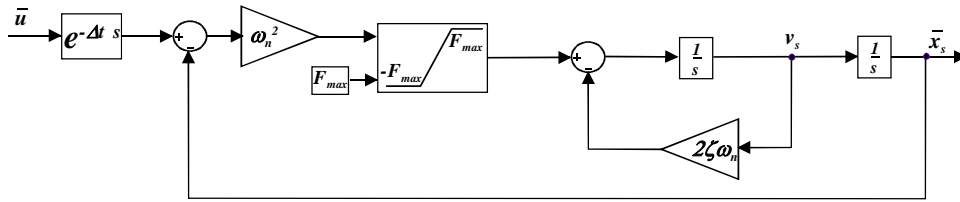


Figura 5.67 Modelo dinâmico com saturação de força,  $F_{max}$ .

## Modelo dinâmico 3

Neste modelo, apresentado na Figura 5.68, testou-se uma saturação de força através do parâmetro,  $F_{max}$ , mas neste caso a limitação é realizada de forma progressiva, com limite em  $p/2 \times F_{max}$ , tendo como base a função arco-tangente ( $atan$ ) com declive unitário na origem. Se se denominar a força de entrada no bloco limitador como  $F_{in}$  e a força de saída do bloco como  $F_{out}$ , então:

$$F_{out} = F_{max} \times atan \frac{F_{in}}{F_{max}} \quad (5.108)$$

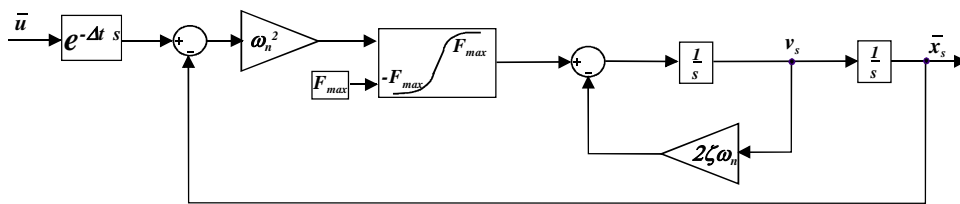


Figura 5.68 Modelo dinâmico com saturação progressiva de força com limite em  $p/2 \times F_{max}$

## Modelo dinâmico 4

Este modelo implementa uma saturação de potência através do parâmetro,  $P_{max}$ . A limitação é realizada em força, sendo o parâmetro  $F_{max}$  obtido através de avaliação contínua da velocidade do órgão móvel da válvula,  $v_s$ , como se ilustra na Figura 5.69.

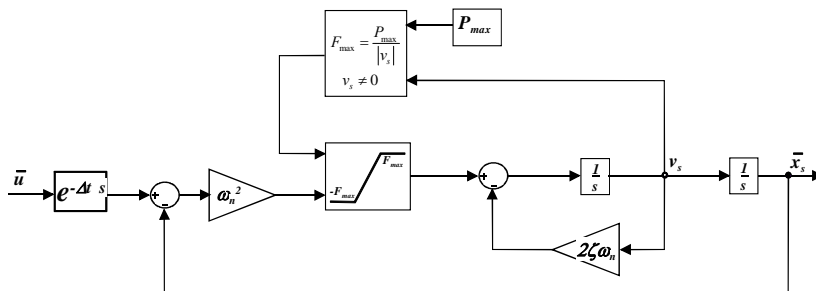


Figura 5.69 Modelo dinâmico com limitação de potência,  $P_{max}$

### Modelo dinâmico 5

Este modelo (Figura 5.70) é idêntico ao anterior residindo a diferença no limitador de força que se baseia na função exponencial e tem uma limitação máxima em  $F_{max} + \Delta F$ . De notar, como se pode inferir de (5.109), que a função limitadora tem derivada contínua para qualquer  $F_{in}$  de entrada.

$$F_{out} = \begin{cases} -F_{max} - DF \times \left(1 - e^{\left(\frac{F_{max} + F_{in}}{DF}\right)}\right) & ; F_{in} \leq -F_{max} \\ F_{in} & ; -F_{max} < F_{in} < F_{max} \\ F_{max} + DF \times \left(1 - e^{\left(\frac{F_{max} - F_{in}}{DF}\right)}\right) & ; F_{in} \geq F_{max} \end{cases} \quad (5.109)$$

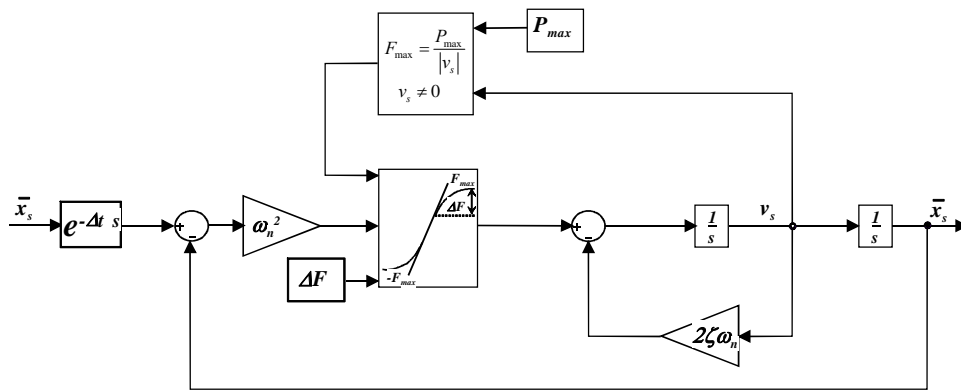


Figura 5.70 Modelo dinâmico com limitação de potência,  $P_{max}$

### Modelo dinâmico 6

Este modelo pode ser considerado um modelo com limitação de força explícita, estando a velocidade também a ser limitada de forma implícita (ver Figura 5.71). Os parâmetros a ajustar são o limite para a força à velocidade nula,  $F_{max}$ , e o limite para a velocidade,  $L_v$ .

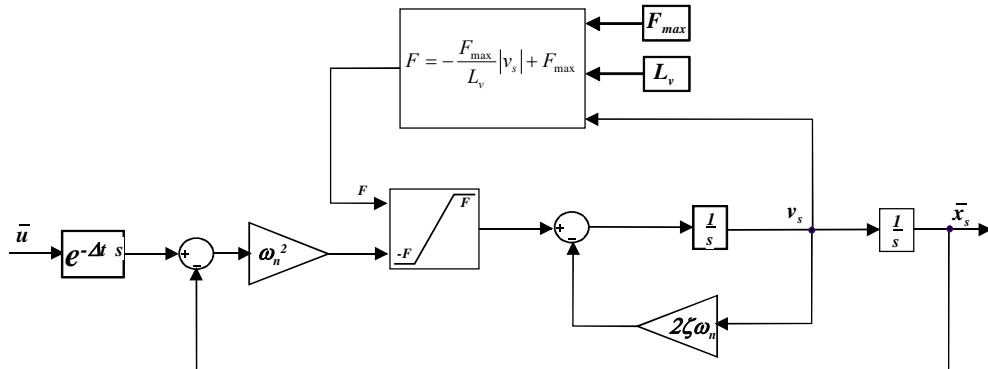


Figura 5.71 Modelo dinâmico com limitação (explícita) de força e limitação (implícita) de velocidade

## Modelo dinâmico 7

Este modelo (Figura 5.72) limita a potência disponível,  $P_{max}$ , a força,  $F_{max}$ , e a velocidade,  $L_v$ .

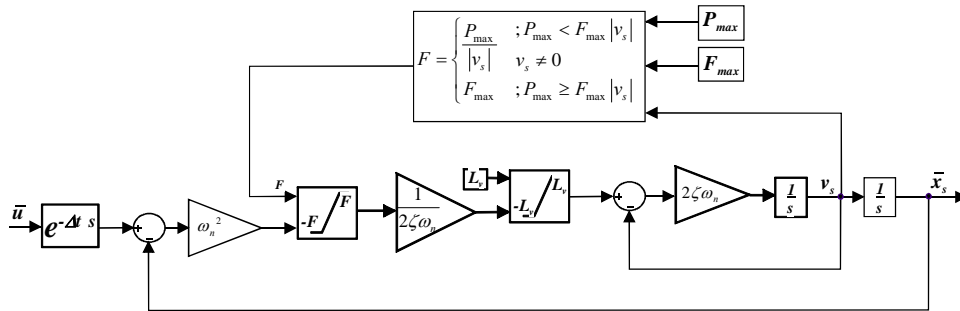


Figura 5.72 Modelo dinâmico com limitação de potência e de força

Na Figura 5.73 ilustram-se os diagramas de Bode para os modelos baseados no limite de força e/ou potência.

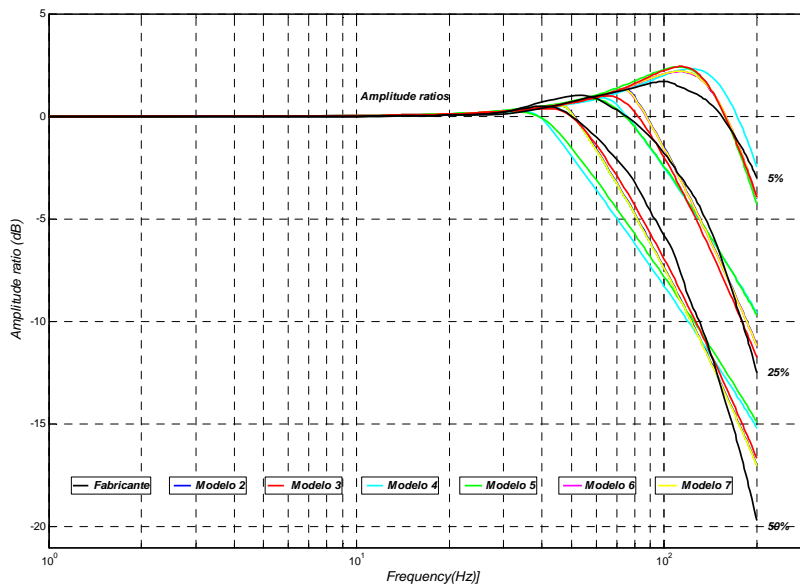


Figura 5.73 Diagramas de Bode (amplitude) para os modelos baseados no limite de força e/ou potência

Modelo 2:  $f_n = 141.0 \text{ s}^{-1}$ ,  $x = 0.430$ ,  $F_{max} = 10.4 \cdot 10^4 \text{ N}$ , ( $F = 18.633$ )

Modelo 3:  $f_n = 152.1 \text{ s}^{-1}$ ,  $x = 0.395$ ,  $F_{max} = 78.9 \cdot 10^3 \text{ N}$ , ( $F = 17.125$ )

Modelo 4:  $f_n = 138.6 \text{ s}^{-1}$ ,  $x = 0.423$ ,  $P_{max} = 77.9 \cdot 10^5 \text{ N s}^{-1}$ , ( $F = 52.695$ )

Modelo 5:  $f_n = 137.8 \text{ s}^{-1}$ ,  $x = 0.417$ ,  $P_{max} = 81.9 \cdot 10^5 \text{ N s}^{-1}$ ,  $DF = 67.4 \cdot 10^3 \text{ N}$ , ( $F = 51.896$ )

Modelo 6:  $f_n = 141.5 \text{ s}^{-1}$ ,  $x = 0.431$ ,  $F_{max} = 10.4 \cdot 10^4 \text{ N}$ ,  $L_v = 46.2 \cdot 10^3 \text{ s}^{-1}$ , ( $F = 18.727$ )

Modelo 7:  $f_n = 141.4 \text{ s}^{-1}$ ,  $x = 0.431$ ,  $F_{max} = 10.5 \cdot 10^4 \text{ N}$ ,  $P_{max} = 77.8 \cdot 10^6 \text{ N s}^{-1}$ ,  $L_v = 135.7 \text{ s}^{-1}$ , ( $F = 17.567$ )

### Modelo dinâmico 8

Este modelo limita a velocidade,  $L_v$ , e a aceleração,  $L_a$ , por saturação abrupta (Figura 5.74).

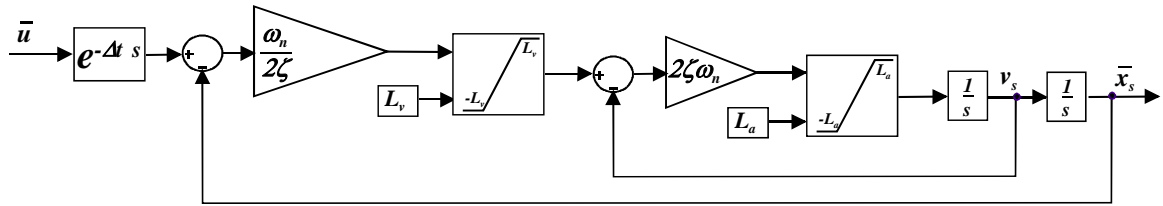


Figura 5.74 Modelo dinâmico com saturação “abrupta” da velocidade e aceleração

### Modelo dinâmico 9

Neste modelo (Figura 5.75), idêntico ao modelo anterior, as limitações de velocidade e aceleração são progressivas com limites em  $p/2 \times L_v$  e  $p/2 \times L_a$ , respectivamente, à custa da função *atan* (de forma semelhante a (5.108)).

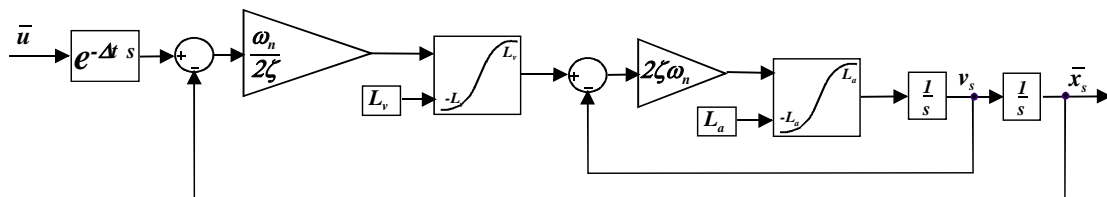


Figura 5.75 Modelo dinâmico com limitação progressiva de velocidade e aceleração

### Modelo dinâmico 10

Modelo com limitação de velocidade,  $L_v + \Delta L_v$ , e limitação de aceleração,  $L_a + \Delta L_a$ , baseado no limitador definido em (5.109), ou seja, através do uso da função exponencial (ver Figura 5.76).

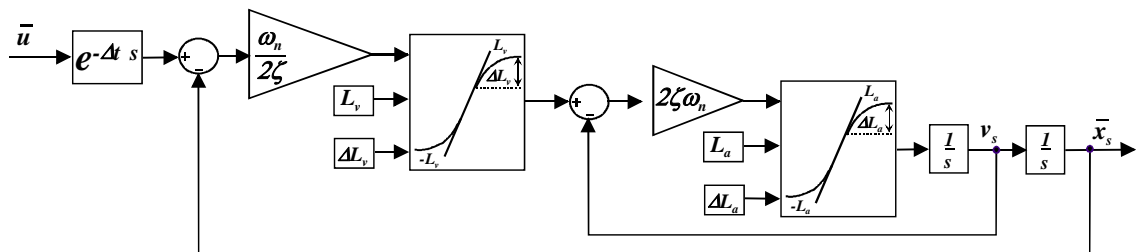


Figura 5.76 Modelo dinâmico com limitação de velocidade e aceleração baseados na função exponencial

## Modelo dinâmico 11

O modelo apresentado na Figura 5.77 possui limitação de velocidade,  $p/2 \times L_v$ , baseado na função  $atan$  (expressão (5.108)) e limitação com saturação abrupta de aceleração,  $L_a$ .

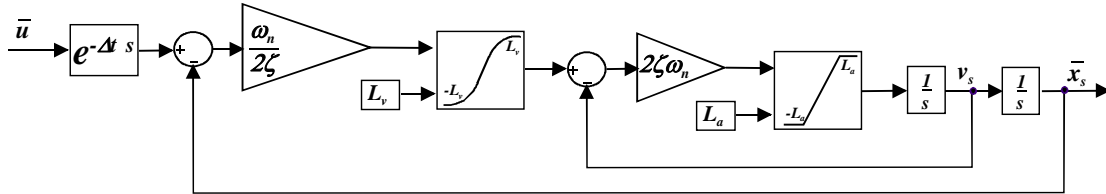


Figura 5.77 Modelo dinâmico com limite progressivo de velocidade e saturação abrupta de aceleração

A figura Figura 5.78 ilustra os resultados obtidos para os modelos 8, 9, 10 e 11.

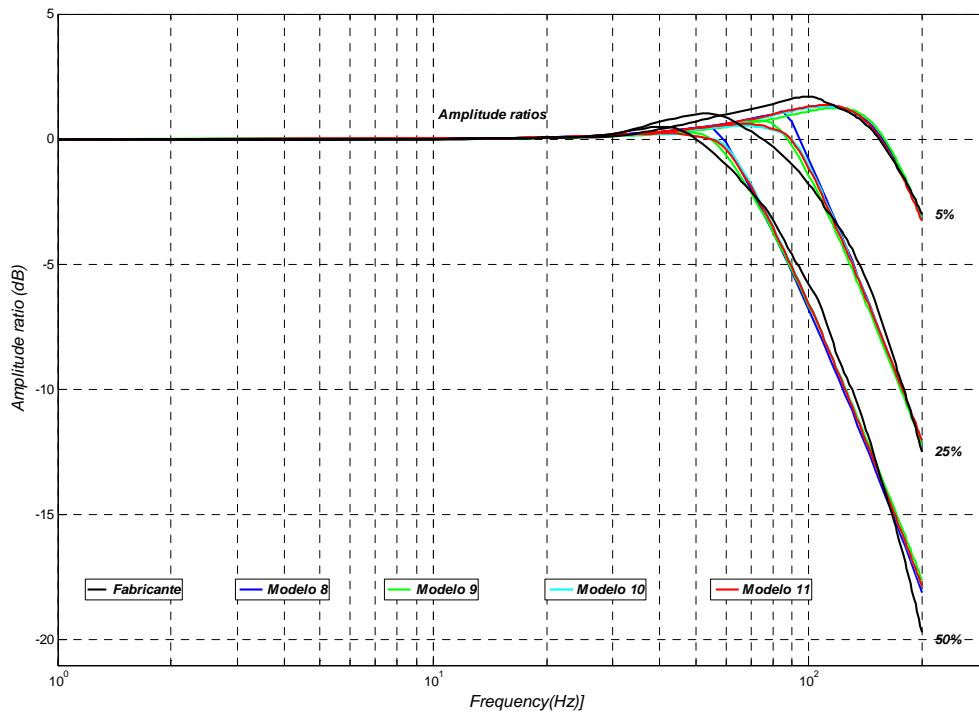


Figura 5.78 Diagramas de Bode (amplitude) para os modelos baseados no limite de velocidade e/ou aceleração

Modelo 8:  $f_n = 154.8 \text{ s}^{-1}$ ,  $x = 0.494$ ,  $L_v = 164.4 \text{ s}^{-1}$ ,  $L_a = 78.3 \cdot 10^3 \text{ s}^{-2}$ , ( $F = 9.056$ )

Modelo 9:  $f_n = 174.2 \text{ s}^{-1}$ ,  $x = 0.500$ ,  $L_v = 119.0 \text{ s}^{-1}$ ,  $L_a = 74.6 \cdot 10^3 \text{ s}^{-2}$ , ( $F = 8.681$ )

Modelo 10:  $f_n = 171.9 \text{ s}^{-1}$ ,  $x = 0.466$ ,  $L_v = 176.7 \text{ s}^{-1}$ ,  $L_a = 81.2 \cdot 10^3 \text{ s}^{-2}$ ,  $DL_v = 175.5 \text{ s}^{-1}$ ,  
 $DL_a = 1.38 \cdot 10^3 \text{ s}^{-2}$ , ( $F = 6.917$ )

Modelo 11:  $f_n = 161.2 \text{ s}^{-1}$ ,  $x = 0.481$ ,  $L_v = 124.9 \text{ s}^{-1}$ ,  $L_a = 81.7 \cdot 10^3 \text{ s}^{-2}$ , ( $F = 6.873$ )

### 5.2.3 Ajuste de fase

Na Figura 5.79 encontram-se os resultados da resposta em frequência - amplitude e fase - resultantes do ajuste dos parâmetros  $(\omega_n, x, L_v, L_a)$ , tendo em conta a amplitude da resposta para o Modelo dinâmico 11 (Figura 5.77). Como se pode ver a fase da resposta necessita de ser melhorada pois a esta tende para  $-180^\circ$  (sistema de 2ª ordem) independentemente da excursão do sinal.

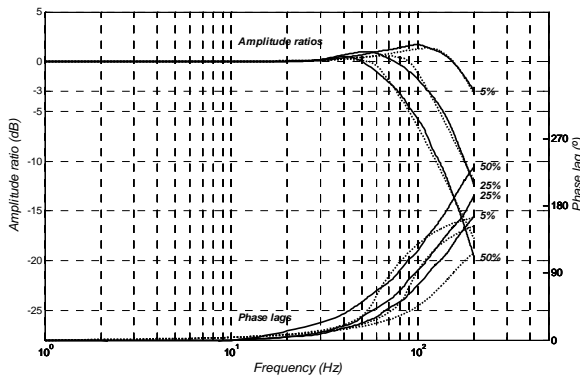


Figura 5.79 Diagrama de Bode (amplitude e fase) para o modelo 11, com

$$f_n = 161.2 \text{ s}^{-1}; x = 0.481; L_v = 124.9 \text{ s}^{-1}; L_a = 81.7 \cdot 10^3 \text{ s}^{-2}; (F = 6.873)$$

A minimização do erro de fase da resposta é feita à custa dum atraso temporal,  $\Delta t$ , à semelhança do proposto em [Almeida93] e [Quintas99]. O método para a identificação deste atraso temporal,  $\Delta t$ , foi idêntico ao usado no ajuste dos outros parâmetros do modelo. O atraso que minimiza o erro de fase, para as diferentes excursões do sinal de entrada, é  $Dt = 7.625 \times 10^{-4} \text{ s}$ . A Figura 5.80 mostra a resposta em frequência do modelo dinâmico 11, até aos 300 Hz.

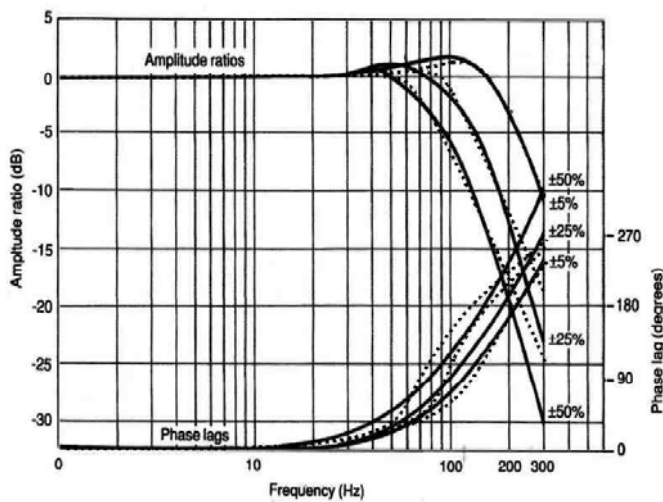


Figura 5.80 Comparação entre os diagramas de Bode simulados (traço interrompido) com os fornecidos pelo fabricante (cheio)

A Figura 5.81 mostra os resultados da parametrização do modelo dinâmico 11 para uma válvula NG6 da Bosch®.  $f_n = 101.4 \text{ s}^{-1}$ ;  $x = 0.55$ ;  $L_v = 271.0 \text{ s}^{-1}$ ;  $L_a = 21.0 \cdot 10^4 \text{ s}^{-2}$ ; ( $F = 2.15$ ).

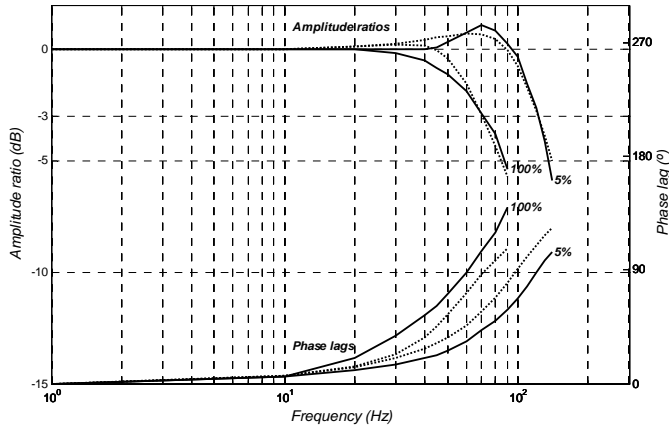


Figura 5.81 Comparação entre os diagramas de Bode simulados (traço interrompido) com os fornecidos pelo fabricante (cheio), para a válvula NG6 da Bosch (ref: 0811404038)

A abordagem seguida, no desenvolvimento dos modelos dinâmicos apresentados, consistiu na utilização de um modelo de 2ª ordem em conjunto com algumas não-linearidades, de modo a aproximar os diagramas de Bode do modelo dos do fornecido pelo fabricante. As não-linearidades inseridas nos modelos procuraram traduzir os efeitos dos limites físicos existentes no movimento da gaveta da válvula. Neste contexto testaram-se diferentes saturações para a limitação de força, de potência, de aceleração e de velocidade. Consideraram-se saturações abruptas e saturações suaves (todas com ganho unitário na origem) para os limitadores. Para as saturações suaves foram utilizadas funções limite baseadas em funções exponenciais, em funções arco-tangente ou em funções do tipo tangente hiperbólica. Na generalidade, os modelos apresentam melhores aproximações quando se utilizam saturações suaves, pois estas estão mais próximas da realidade física. As exceções detectadas foram na saturação de aceleração onde as saturações abruptas têm desempenho idêntico. A variedade de modelos desenvolvidos permitiu avaliar os melhores desempenhos, de sistemas de 2ª ordem, para a modelação da parte dinâmica da válvula. A extrapolação dos resultados, aqui apresentados para o modelo KBSDG4V-3 da Vickers, permite destacar alguns modelos, uns pelo compromisso simplicidade/desempenho e, outros, pelo seu desempenho. Pela relação simplicidade/desempenho destacam-se, com desempenho idêntico, o modelo 1, Figura 5.66, ( $F=15.982$ ), saturação suave de velocidade, e o modelo 3, Figura 5.73, ( $F=17.125$ ), saturação suave de força. Avaliando somente o desempenho destaca-se o modelo dinâmico 11 que apresenta uma limitação de velocidade baseada na função arco-tangente e uma saturação abrupta da aceleração; é interessante notar que o melhor limitador de velocidade está em



acção desde as baixas velocidades, mas que o limite de aceleração tem melhores resultados quando se utiliza uma saturação abrupta.

Quanto à resposta em fase pode concluir-se, a partir da Figura 5.80, que um bom resultado é obtido para a característica de 5% de amplitude (dentro da frequência de utilização da válvula). Contudo, as outras características de fase mostram que o modelo pode ser melhorado, especialmente nas frequências mais elevadas. Os resultados da simulação do modelo dinâmico 11 mostram que os artefactos das dinâmicas não modeladas são mais acentuados nas frequências acima dos 200Hz. Na sequência dos resultados obtidos com um modelo de 2ª ordem, justifica-se uma avaliação futura de modelos de 3ª ou de ordem superior. Uma outra direcção possível, e tomando o exemplo da válvula NG6 da Bosch (Figura 5.81), consiste em fazer variar o coeficiente de amortecimento,  $x$ , por exemplo, com a velocidade.

## 6 EXPERIÊNCIAS DE SIMULAÇÃO COM *HARDWARE-IN-THE-LOOP*

Neste capítulo apresentam-se a plataforma experimental desenvolvida, a parametrização de alguns modelos de componentes hidráulicos e algumas experiências com *hardware-in-the-loop* realizadas.

### 6.1 Plataforma experimental

A necessidade de verificação e parametrização de modelos de componentes hidráulicos levou ao projecto de uma plataforma experimental. Esta plataforma tentou servir outros propósitos não só na área da modelação e simulação, mas também na área do controlo de sistemas hidráulicos. Para estes fins foi projectado e desenvolvido um manipulador de três eixos do tipo SCARA (*Selective Compliant Assembly Robot Arm*) de actuação hidráulica. A Figura 6.1 e a Figura 6.2 ilustram o desenho 3D do manipulador.

#### Projecto mecânico e actuação

Optou-se por uma construção modular baseada em perfis de alumínio de elevada rigidez, com uma coluna a suportar as partes móveis do manipulador. As bases para as válvulas de controlo foram maquinadas em duralumínio (adequado ao fabrico de moldes). Estas opções construtivas permitiram reduzir a massa móvel do manipulador, não comprometendo a sua rigidez.

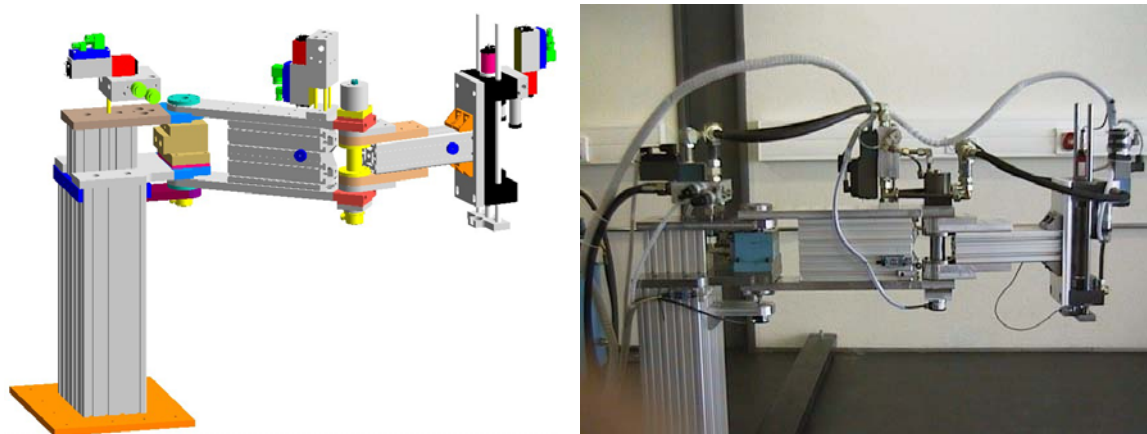


Figura 6.1 Desenho 3D e fotografia do manipulador hidráulico

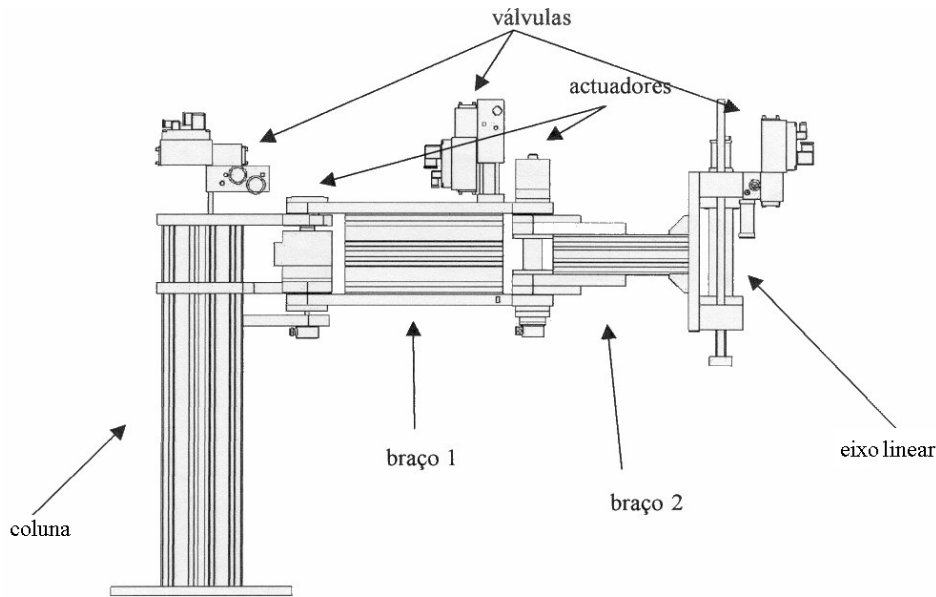


Figura 6.2 Desenho 3D do manipulador hidráulico

A actuação do manipulador é totalmente hidráulica e é realizada, no braço 1, por um actuador de palheta simples, modelo MP-32 da TEXTRON, com um curso de  $280^\circ$  e um binário de 180Nm a uma pressão de 70bar; no braço 2, a actuação fica a cargo de um actuador de características idênticas, modelo MP-22, mas com um binário menor, 75Nm; o eixo linear é construído com um cilindro hidráulico de 40mm de diâmetro e 200mm de curso, com a particularidade de possuir vedantes hidrodinâmicos de baixo atrito. Os actuadores são comandados por servo-válvulas distribuidoras proporcionais com posição de segurança fechada. O circuito hidráulico do manipulador encontra-se na Figura 6.3.

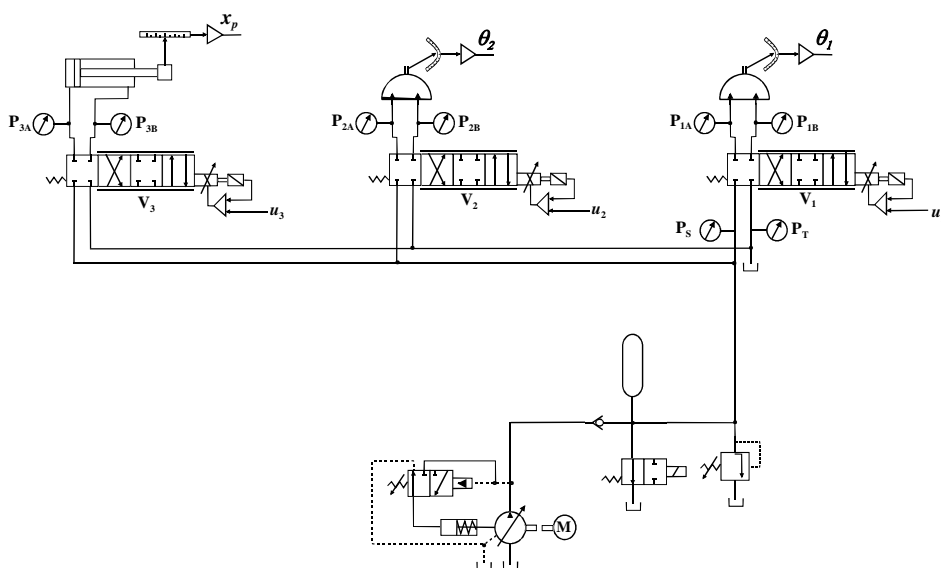


Figura 6.3 Circuito hidráulico do manipulador

Os actuadores dos braços 1 e 2 são comandados por válvulas NG6 da Bosch com caudal nominal de 24 e 12l/min, respectivamente; o actuador do eixo linear recorre a uma válvula da Vickers, modelo KBSDG4V-3, com um caudal nominal de 24l/min. A central hidráulica está habilitada a debitar caudais até 15l/min, com a pressão máxima de 200 bar. Utilizou-se uma bomba de pistões axiais de cilindrada variável PVQ10 da Vickers, em conjunto com um acumulador IVH 5-330 da OLAER, com 5l de volume, de modo a obter uma pressão constante para alimentação do circuito hidráulico (Figura 6.3).

### **Instrumentação do manipulador**

Os objectivos a satisfazer com a plataforma experimental levaram à instalação de transdutores para medição das várias grandezas, não só necessárias ao seu controlo, mas também à parametrização dos modelos. Nesse sentido foram instalados transdutores para medição das posições (angular e linear) e aceleração instantânea em todos os braços do manipulador e, também, fins de curso eléctricos tendo em vista a indexação automática dos codificadores angulares usados na medição da posição angular das juntas rotativas. Nestas medições usaram-se codificadores angulares incrementais, acoplados directamente aos veios dos actuadores, tipo 2RK5000-S da SCANCON, com uma resolução de  $1.257 \times 10^{-3}$  rad. A posição do eixo linear é obtida a partir de um transdutor de posição linear (MTS-Temposonics III), com uma gama de 0-200mm e saída em corrente (4-20mA). Instalaram-se também transdutores de pressão de modo a monitorar a pressão relativa em todas as câmaras dos actuadores, na bomba e no tanque; os transdutores de pressão são da HBM (P15RVA1) com saída em tensão de 0-10V, para uma gama de pressões de 0-200 bar, largura de banda de 2kHz e não linearidade máxima de 1%. As acelerações instantâneas nos três eixos são medidas através de acelerómetros da FGP Instrumentation (FA 201), com uma gama de  $\pm 5g$ , saída em tensão de  $\pm 10V$ , largura de banda de 250Hz e não linearidade máxima de 2%. Além dos transdutores acima referidos, foi também integrado no sistema de aquisição de dados um transdutor de caudal da Webster Instruments, tendo sido desenvolvida toda a electrónica de interface com as cartas de controlo e aquisição.

### **Hardware e software para controlo e aquisição de dados**

Para a aquisição de dados e controlo do manipulador foram usadas uma carta da National Instruments (PCI 1200) e uma carta da dSPACE (DS1102), sincronizadas através de uma linha digital, *SincAq*, como se ilustra na Figura 6.4.

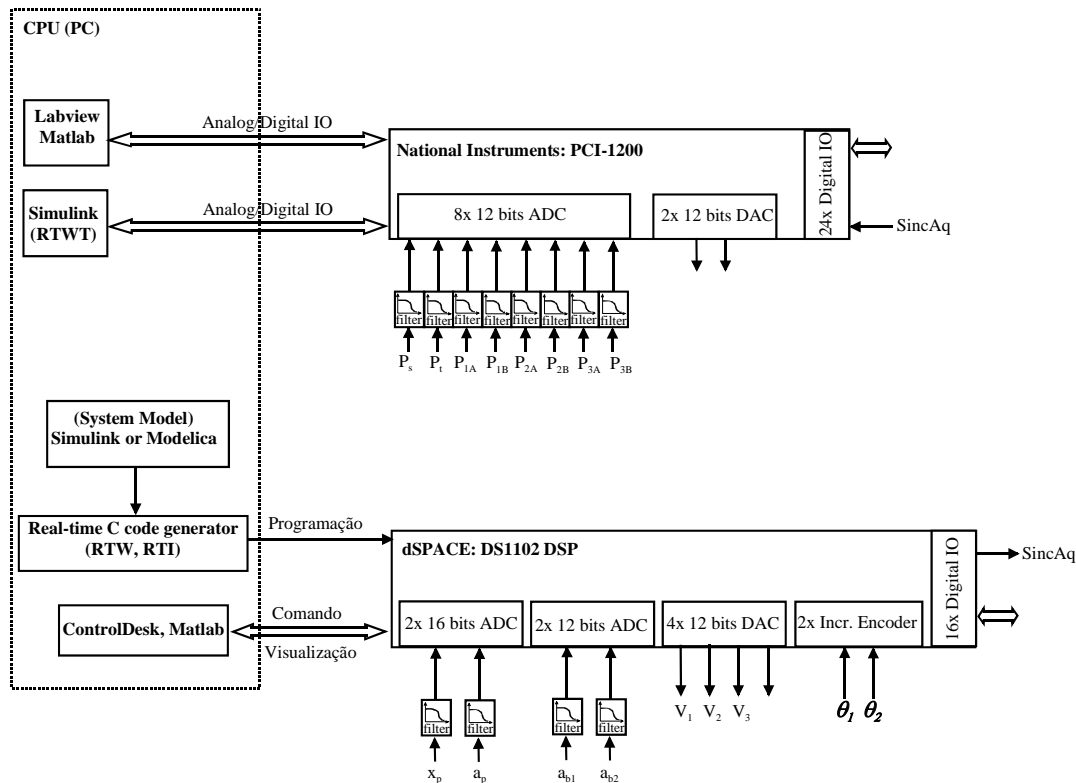


Figura 6.4 Placas de hardware e software utilizado

A carta PCI 1200 é uma carta de baixo custo que possui oito canais de entrada analógicos com uma resolução de 12 bits, dois canais de saída analógicos e 24 linhas de I/O digital. Esta carta está mais vocacionada para a implementação de sistemas de aquisição de dados, muito embora possa também ser usada para controlo, através da utilização do Simulink e de um núcleo de tempo real fornecido pela Mathworks (*Real Time Windows Target – RTWT*). A sua operação em modo de aquisição pode ser feita através do LabView ou do Matlab, usando para isso a *Data Aquisition Toolbox*.

A carta DS1102 da dSPACE, construída com a DSP TMS320C31 da Texas Instruments, também uma carta de baixo custo, está vocacionada para simulação em tempo real de sistemas (contínuos e/ou discretos) e, portanto, para a implementação de controladores, já que garante a execução dos algoritmos de controlo em tempo real. Possui duas entradas analógicas de 16 bits e duas de 12 bits, quatro saídas analógicas de 12 bits, 16 linhas de I/O digital e duas entradas para ligação de codificadores incrementais. Os modelos para simulação e/ou os controladores podem ser implementados, na presente plataforma, em Simulink ou Modelica, sendo o código de tempo real gerado em linguagem C, recorrendo ao *Real Time Workshop (RTW)* da Mathworks e ao *Real Time Interface (RTI)* da dSPACE. A interação com o programa pode ser realizada através da aplicação *ControlDesk* da dSPACE ou através do Matlab usando, para isso, a biblioteca de funções *MLIB* fornecida pela dSPACE. As bases de hardware e software permitem a flexibilidade necessária para

a utilização da plataforma experimental em tarefas de controlo, e também na área da modelação e simulação, já que é possível obter, com alguma facilidade, medidas das grandezas necessárias à parametrização e validação dos modelos desenvolvidos. Além disso, a plataforma foi também usada em tarefas de ensino nas áreas do controlo automático e da modelação e simulação de sistemas físicos.

### Software desenvolvido para interacção com o manipulador

Foi desenvolvido software de apoio para a experimentação de diferentes algoritmos de controlo e para a aquisição dos dados necessários à parametrização dos modelos dos componentes hidráulicos. Todo o software de controlo do manipulador, incluindo indexação automática e geração de trajectórias ponto a ponto, foi desenvolvido em Simulink, sendo posteriormente traduzido para linguagem C e instalado na carta DS1102 através de software dedicado da dSPACE. Foi também desenvolvido software de monitorização e operação em tempo real do manipulador, usando a plataforma ControlDesk da dSPACE (Figura 6.5 e Figura 6.6). Por operação em tempo real entende-se, por exemplo, a variação dos parâmetros do controlador ou a operação manual do manipulador.

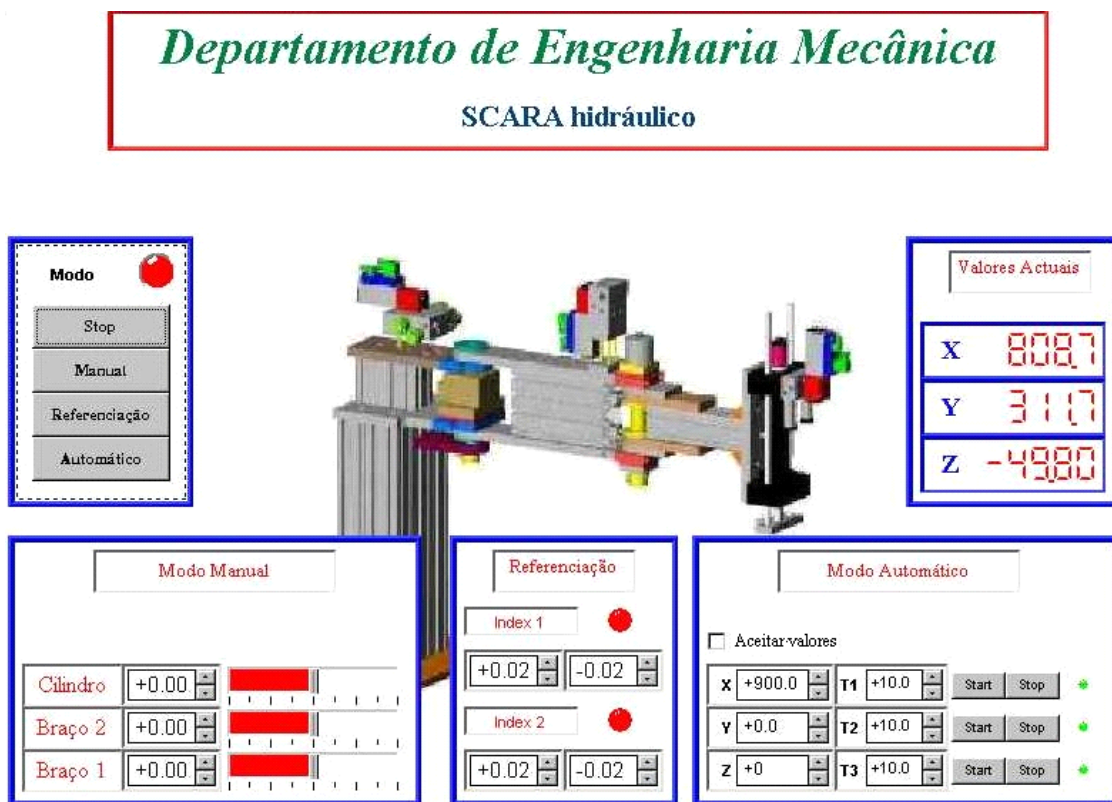


Figura 6.5 Janela principal da aplicação desenvolvida para o comando do manipulador

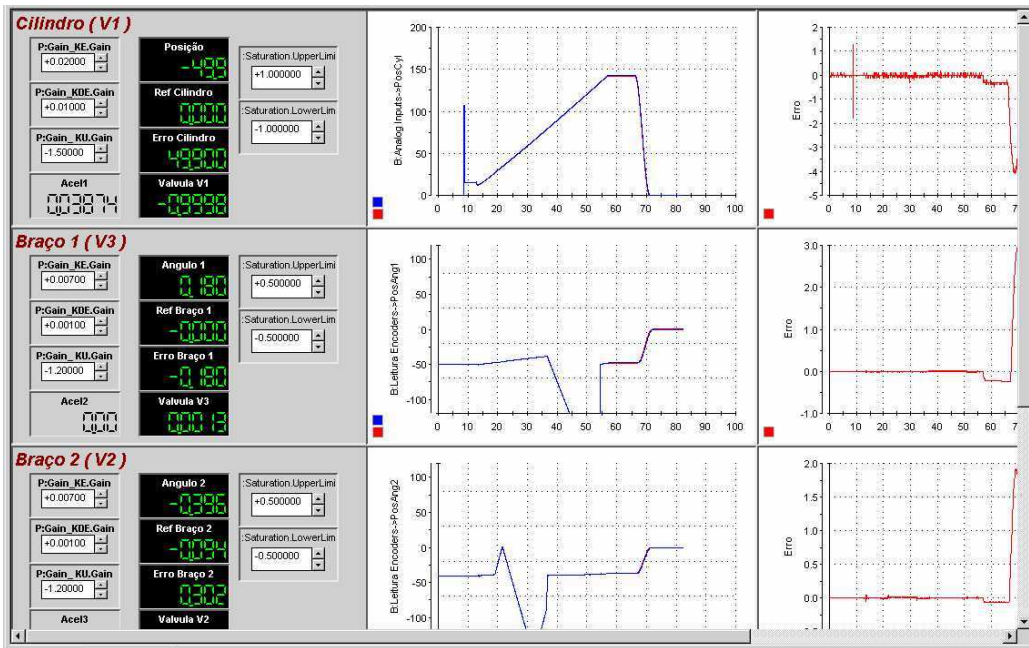


Figura 6.6 Janela da página elaborada para aquisição de dados

Para a aquisição de dados dos transdutores de pressão desenvolveu-se uma aplicação na plataforma LabView (ver Figura 6.9) com possibilidade de sincronização com o software de controlo, através de canais digitais da carta de aquisição de dados e da carta de controlo.

A organização dos blocos para a implementação do controlador é apresentada nas figuras 6.7 e 6.8. Note-se que o bloco “Controladores das juntas” é facilmente re-configurável para o uso de diferentes algoritmos de controlo. Na presente versão são usados controladores de lógica difusa (projecto elaborado em colaboração com o Departamento de Engenharia Electrónica e Telecomunicações da Universidade de Aveiro, no âmbito da disciplina de Projecto, [Fonseca00]).

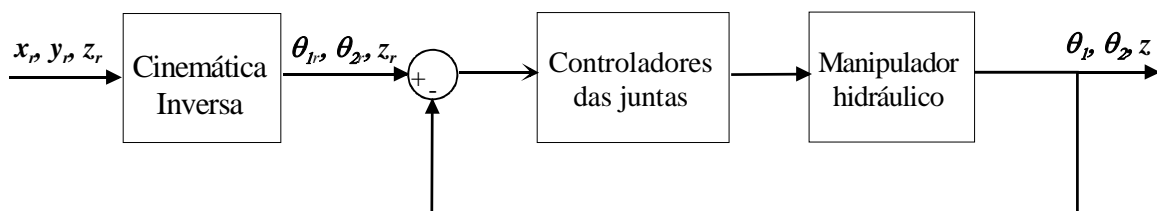


Figura 6.7 Diagrama de blocos do controlador do manipulador

O manipulador SCARA, para além de ser utilizado como instrumento para a parametrização de modelos e de avaliação do desempenho de algoritmos de controlo em sistemas de actuação hidráulica, também tem vindo a ser usado como plataforma experimental em várias disciplinas da Licenciatura de Engenharia Mecânica da Universidade de Aveiro. Mais recentemente, o eixo linear

foi usado como o cilindro de trabalho de uma “mini-prensa” hidráulica de 20 KN de capacidade à compressão e 14 KN à tracção (para uma pressão de trabalho de 160bar). Foram já realizados trabalhos de teste à fadiga de amortecedores [Talaia02] e estão em curso trabalhos de teste à fadiga de próteses da anca (Figura 6.10).

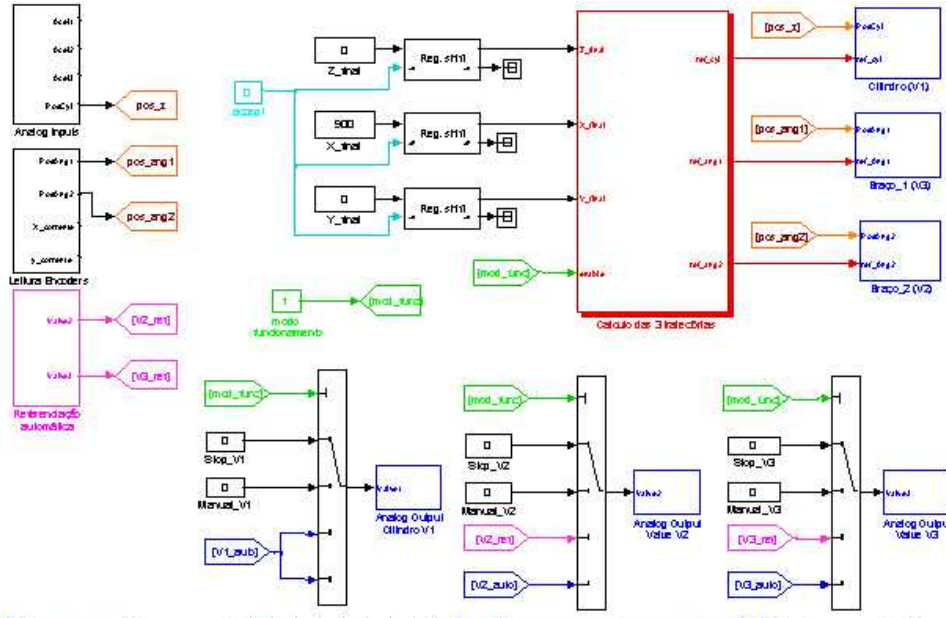


Figura 6.8 Diagrama de blocos em Simulink da aplicação de controlo e monitorização do manipulador

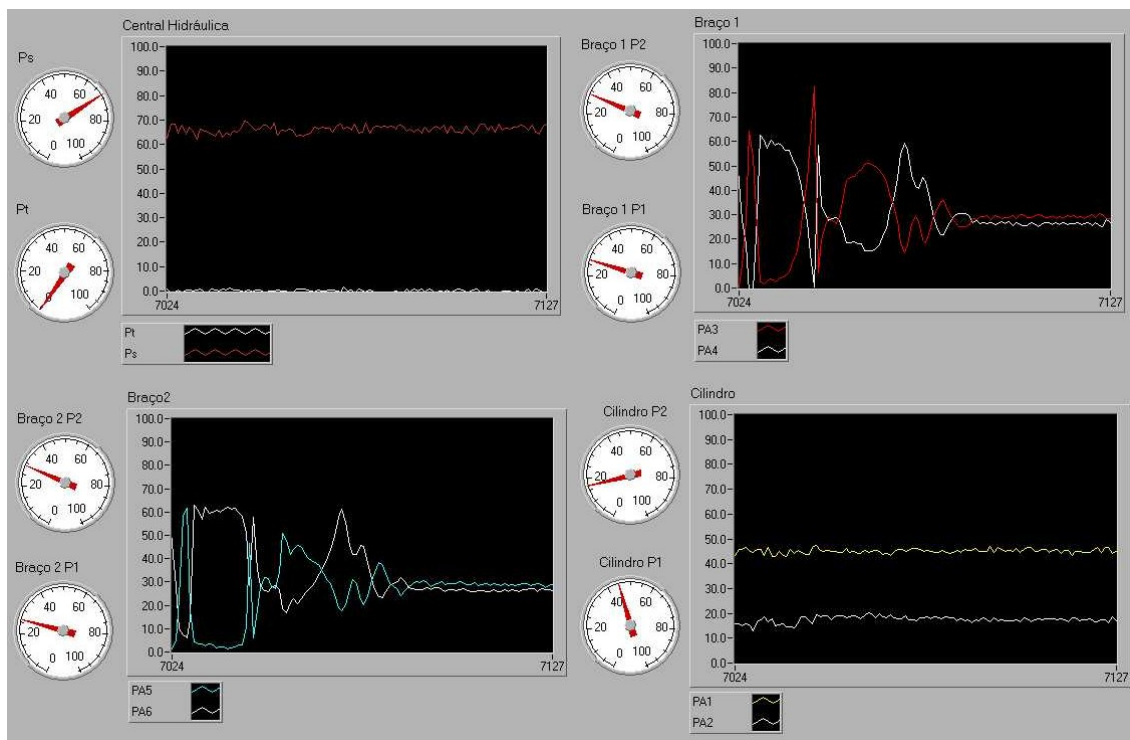


Figura 6.9 Aplicação em LabView para monitorização das pressões nas câmaras dos actuadores do SCARA



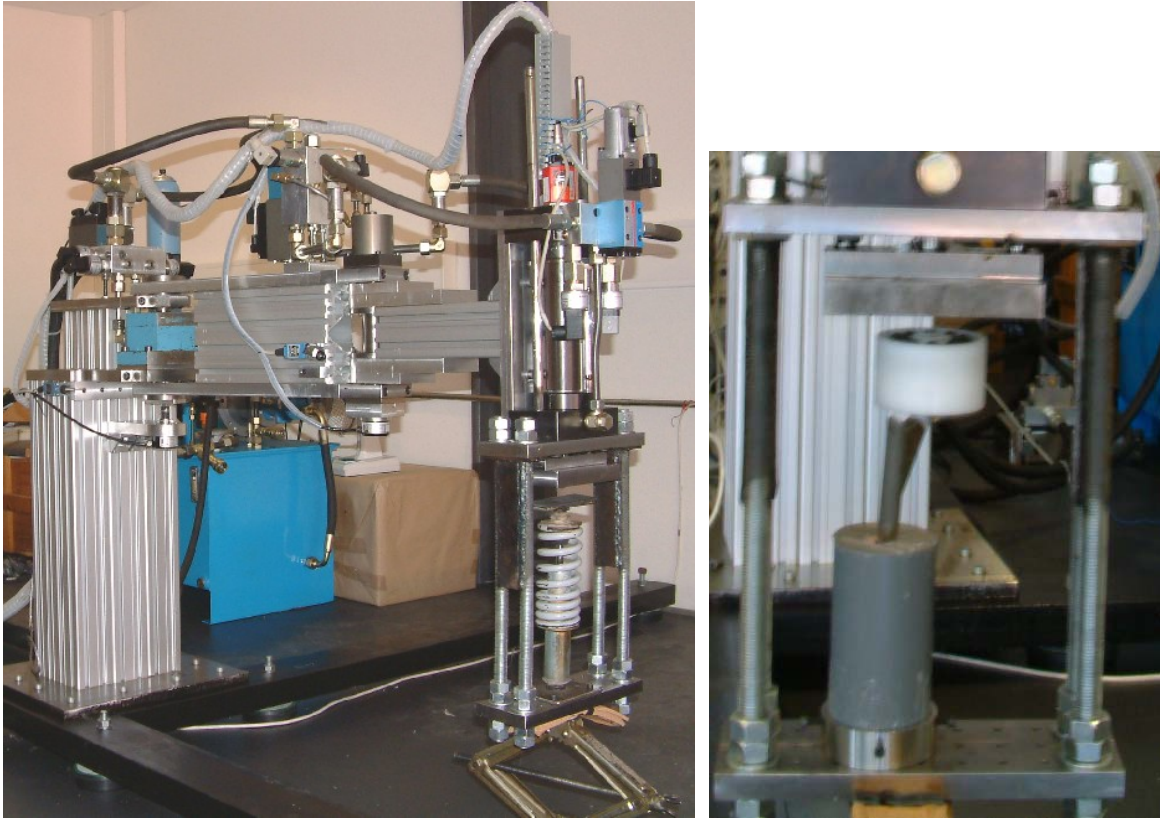


Figura 6.10 Estrutura para adaptação do eixo linear para realização de ensaios à tracção e compressão (ensaio de fadiga em amortecedor e ensaios em prótese de anca)

## 6.2 Identificação de parâmetros de alguns modelos

Nesta secção é proposto um conjunto de experiências para a determinação dos parâmetros de alguns modelos, apresentados nos Capítulos 4 e 5, tendo em vista o seu uso na realização de experiências com *hardware-in-the-loop*.

### 6.2.1 Medição indirecta do módulo de compressibilidade

O módulo de compressibilidade efectivo,  $\beta_e$ , foi estimado através da comparação da aceleração máxima do êmbolo de um cilindro hidráulico, e da frequência à qual esta ocorre, com os resultados da simulação de um modelo linear do conjunto válvula+cilindro.

A plataforma usada para a parametrização dos modelos para o módulo de compressibilidade efectivo é apresentada na Figura 6.11. O diagrama correspondente do conjunto válvula + cilindro mostra-se na Figura 6.12.

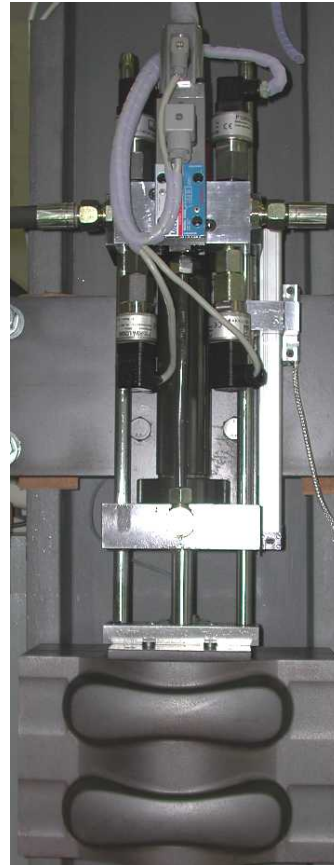
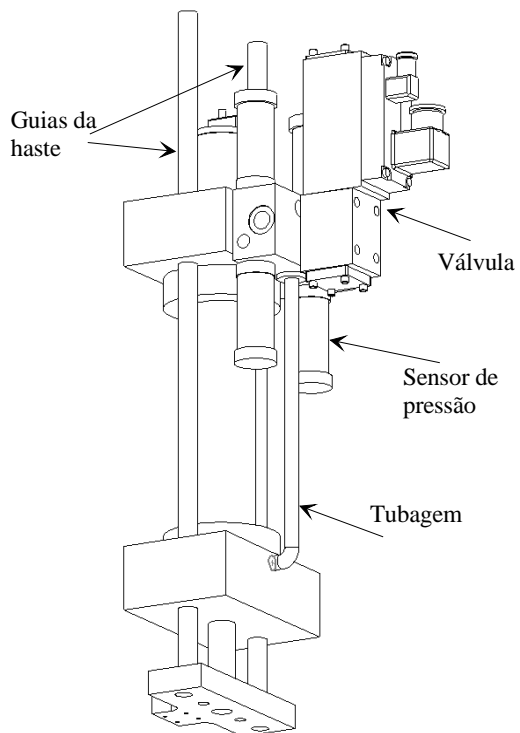


Figura 6.11 Plataforma hidráulica (desenho 3D e sistema real).

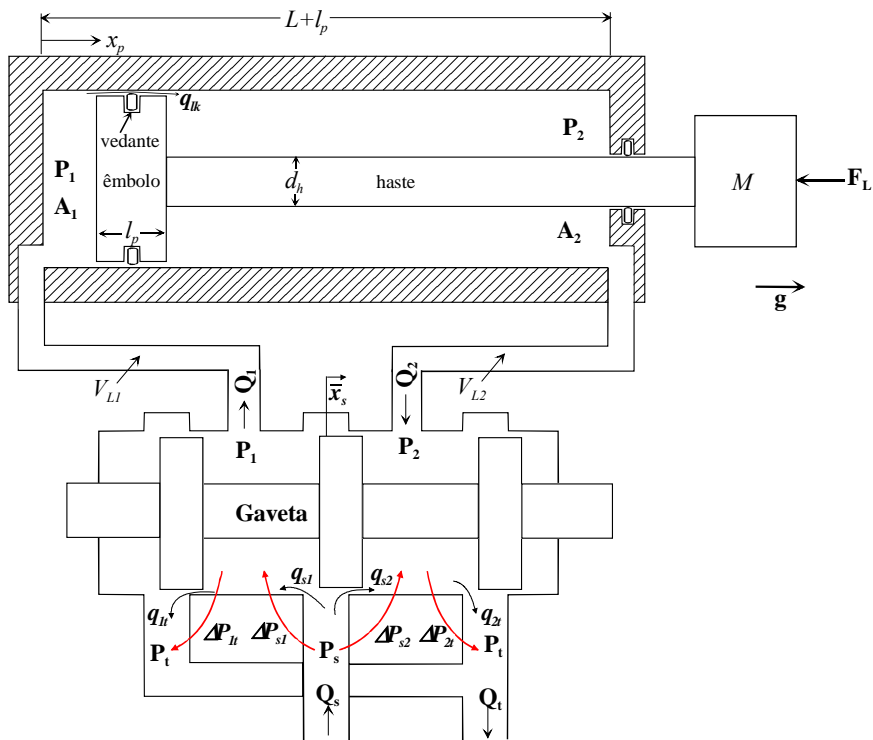


Figura 6.12 Diagrama da plataforma usada para a medição do módulo de compressibilidade efectivo

### 6.2.1.1 Modelo linear do conjunto válvula + cilindro

Numa situação de equilíbrio na posição do êmbolo,  $X_{p0}$ , com velocidade e aceleração nulas, a força resultante é nula e, então,  $A_1 P_{10} + M \times g - A_2 P_{20} = 0$ . Da expressão  $P_s = P_1 + P_2$  (5.7), e para as pressões de equilíbrio tem-se  $P_{20} = P_s - P_{10}$ . Então:

$$P_{10} = \frac{A_2 P_s - M \times g}{A_1 + A_2} \quad (6.1)$$

$$P_{20} = \frac{A_1 P_s - M \times g}{A_1 + A_2} \quad (6.2)$$

O sistema foi linearizado em torno de  $\begin{matrix} \bar{P}_{10} & \bar{P}_{20} & \bar{V}_{p0} & \bar{X}_{p0} & \bar{X}_{s0} \end{matrix}$ , com  $\bar{V}_{p0} = 0$  e  $\bar{X}_{s0}$  definido de modo a que as pressões nas câmaras do cilindro sejam iguais a  $P_{10}$  e  $P_{20}$ . Os valores de  $\bar{X}_{s0}$  podem ser obtidos a partir das curvas do ganho de pressão da válvula (apresentados na secção 5.1). Para uma massa em movimento  $M = 80Kg$ , pressão de alimentação  $P_s = 70bar$  e as duas áreas das câmaras  $A_1 = 1.2566 \times 10^{-3} m^2$  e  $A_2 = 8.7650 \times 10^{-4} m^2$ , essas pressões de equilíbrio ocorrem para  $\bar{X}_{s0} = 0.009$ . Os caudais de equilíbrio  $Q_{10}$  e  $Q_{20}$  também são praticamente nulos se se desprezarem as fugas internas do cilindro. As equações linearizadas para os caudais são:

$$\begin{cases} dq_1 = \frac{\partial Q_1}{\partial \bar{x}_s} d\bar{x}_s - \frac{\partial Q_1}{\partial P_1} dp_1 \\ dq_2 = \frac{\partial Q_2}{\partial \bar{x}_s} d\bar{x}_s - \frac{\partial Q_2}{\partial P_2} dp_2 \end{cases} \quad (6.3)$$

$$\text{com } \frac{\partial Q_1}{\partial P_1} = \frac{\partial Q_1 / \partial \bar{x}_s}{\partial P_1 / \partial \bar{x}_s} \text{ e } \frac{\partial Q_2}{\partial P_2} = \frac{\partial Q_2 / \partial \bar{x}_s}{\partial P_2 / \partial \bar{x}_s}$$

Os ganhos de caudal e de pressão são definidos por:

$$k_{q1} = \left. \frac{\partial Q_1}{\partial \bar{x}_s} \right|_{P_{10}}, \quad k_{q2} = \left. \frac{\partial Q_2}{\partial \bar{x}_s} \right|_{P_{20}}, \quad k_{p1} = \left. \frac{\partial P_1}{\partial \bar{x}_s} \right|_{\bar{X}_{s0}}, \quad k_{p2} = \left. \frac{\partial P_2}{\partial \bar{x}_s} \right|_{\bar{X}_{s0}} \quad (6.4)$$

onde  $k_{q1}$  é o ganho de caudal  $Q_1$  em  $\bar{X}_{s0}$ ,  $k_{q2}$  é o ganho de caudal  $Q_2$  em  $\bar{X}_{s0}$ ,  $k_{p1}$  é o ganho de pressão na câmara 1 da válvula para  $\bar{X}_{s0}$  e  $k_{p2}$  é o ganho de pressão na câmara 2 da válvula para  $\bar{X}_{s0}$ . As equações lineares para os caudais podem então ser expressas da forma seguinte:

$$\begin{cases} dq_1 = k_{q1} \times d\bar{x}_s - \frac{k_{q1}}{k_{p1}} dp_1 \\ dq_2 = k_{q2} \times d\bar{x}_s - \frac{k_{q2}}{k_{p2}} dp_2 \end{cases} \quad (6.5)$$

Note-se que  $\frac{k_{q2}}{k_{p2}}$  tem um valor negativo e  $\frac{k_{q1}}{k_{p1}}$  tem um valor positivo. Considerem-se os coeficientes caudal-pressão:

$$\begin{aligned} k_{c1} &= \frac{k_{q1}}{k_{p1}} \\ k_{c2} &= -\frac{k_{q2}}{k_{p2}} \end{aligned} \quad (6.6)$$

deste modo o sistema (6.5) pode ser expresso por

$$\begin{aligned} dq_1 &= k_{q1} \times \bar{x}_s - k_{c1} \times dp_1 \\ dq_2 &= k_{q2} \times \bar{x}_s + k_{c2} \times dp_2 \end{aligned} \quad (6.7)$$

A substituição dos caudais definidos por (6.7) nas equações dinâmicas das pressões nas câmaras do cilindro, em conjunto com a equação linear do movimento (ver secção 4.2), e considerando os caudais de fugas do cilindro nulas, permite estabelecer as equações linearizadas do conjunto válvula e cilindro:

$$\begin{aligned} dp_1 &= \frac{b_{e1}}{V_1} (K_{q1} \times d\bar{x}_s - K_{c1} \times dp_1 - A_1 \times dv_p) \\ dp_2 &= \frac{b_{e2}}{V_2} (-K_{q2} \times d\bar{x}_s - K_{c2} \times dp_2 + A_2 \times dv_p) \\ d\dot{x}_p &= \frac{1}{M} \times (A_1 \times dp_1 - A_2 \times dp_2 + M \times g - f \times dv_p) \\ d\dot{x}_p &= dv_p \end{aligned} \quad (6.8)$$

onde  $V_1 = V_{01} + A_1 \times X_{p0}$  e  $V_2 = V_{02} + A_2 \times (L - X_{p0})$  representam os volumes mortos, condutas e câmaras da válvula, e  $M$  representa toda a massa em movimento (êmbolo, haste, guias e massa movida).

A versão linearizada do modelo de atrito de LuGre (ver secção 4.3.2) é apenas válida para pequenos deslocamentos ( $<15 \mu m$ ), onde a deformação do vedante (variável  $z$ ) é igual ao deslocamento do êmbolo ( $x_p$ ), isto é, o modelo linear só é válido no estado de colagem. O atrito viscoso  $f$  é usado para modelar todos os efeitos do atrito (válvula e cilindro) que ocorrem quando o sinal da velocidade tem variações bastante rápidas. Não se consideram as fugas internas do cilindro e a movimentação do cilindro é feita na vertical. As equações linearizadas podem ser expressas através do espaço de estados seguinte:

$$\begin{aligned}
 \begin{pmatrix} \dot{\delta p}_1 \\ \dot{\delta p}_2 \\ \dot{\delta v}_p \\ \dot{\delta x}_p \end{pmatrix} &= \begin{pmatrix} \frac{b_{e1}}{V_1} K_{c1} & 0 & -\frac{b_{e1}}{V_1} A_1 & 0 \\ 0 & -\frac{b_{e2}}{V_2} K_{c2} & \frac{b_{e2}}{V_2} A_2 & 0 \\ \frac{A_1}{M} & -\frac{A_2}{M} & -\frac{f}{M} & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} \delta p_1 \\ \delta p_2 \\ \delta v_p \\ \delta x_p \end{pmatrix} + \begin{pmatrix} \frac{b_{e1}}{V_1} K_{q1} & 0 \\ \frac{b_{e2}}{V_2} K_{q2} & 0 \\ 0 & 1 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \delta x_s \\ \delta x_g \end{pmatrix} \\
 \begin{pmatrix} \delta v_p \\ \delta x_p \end{pmatrix} &= \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \delta p_1 \\ \delta p_2 \\ \delta v_p \\ \delta x_p \end{pmatrix}
 \end{aligned} \tag{6.9}$$

A função de transferência, que relaciona a posição do cilindro,  $x_p$ , com o deslocamento  $x_s$  da gaveta da válvula, é muito comum na explicitação do comportamento linear do conjunto válvula+cilindro. O modelo linear descrito pelo espaço de estados (6.9) pode ser representado pelo diagrama de blocos da Figura 6.13.

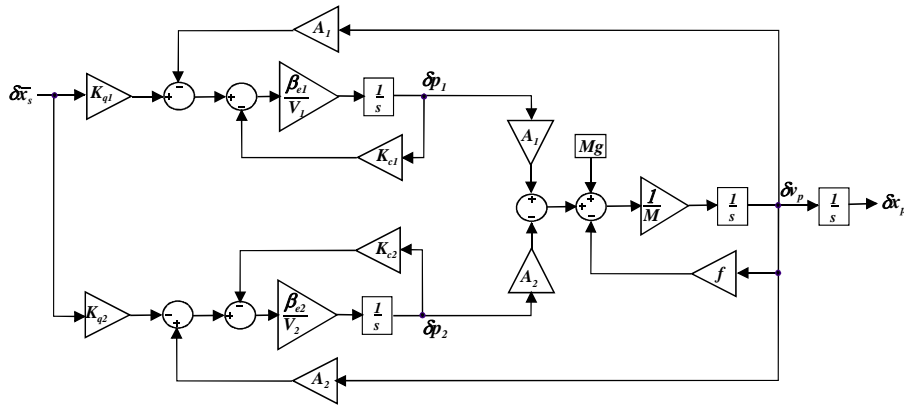


Figura 6.13 Diagrama de blocos correspondente ao espaço de estados (6.9)

Invertendo os sinais nos somadores e simplificando tem-se o diagrama da Figura 6.14:

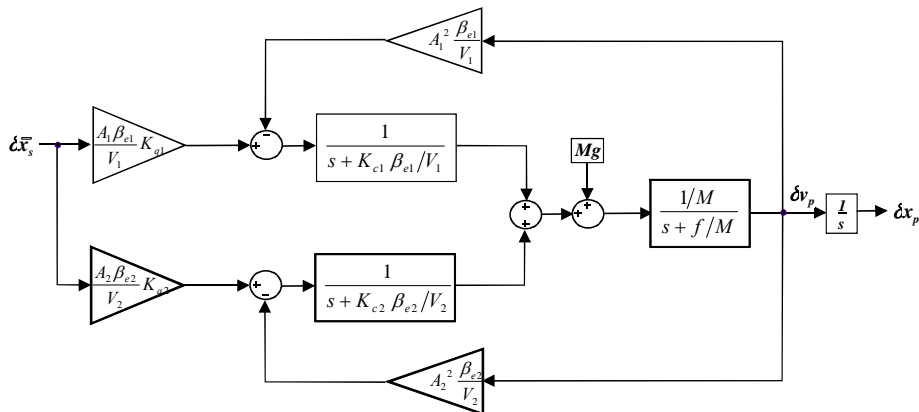


Figura 6.14 Diagrama de blocos simplificado correspondente ao espaço de estados (6.9)

Para reduzir o modelo utiliza-se a aproximação usada por Quintas, em [Quintas99], que considera uma constante de tempo média definida por:

$$\frac{1}{t_m} = \frac{1}{2} \left( \frac{1}{t_1} + \frac{1}{t_2} \right) \quad (6.10)$$

onde  $\frac{1}{t_1} = \frac{K_{c1} b_{e1}}{V_1}$  e  $\frac{1}{t_2} = \frac{K_{c2} b_{e2}}{V_2}$ . Considera-se também um valor intermédio para o módulo de compressibilidade efectivo:  $b_e = \frac{b_{e1} + b_{e2}}{2}$ . Tem-se então

$$\frac{1}{t_m} = \frac{b_e}{2} \left( \frac{K_{c1}}{V_1} + \frac{K_{c2}}{V_2} \right) \quad (6.11)$$

O diagrama de blocos da Figura 6.14 pode então simplificar-se, sendo o resultado da simplificação apresentado nas duas figuras 6.15 e 6.16.

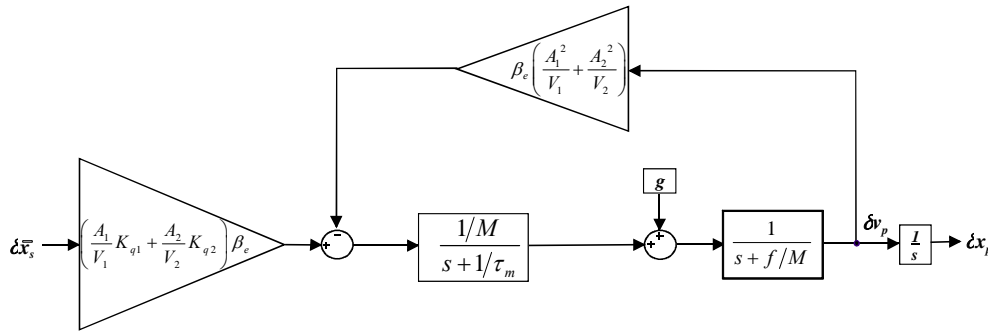


Figura 6.15 Diagrama de blocos simplificado usando (6.10) e (6.11)

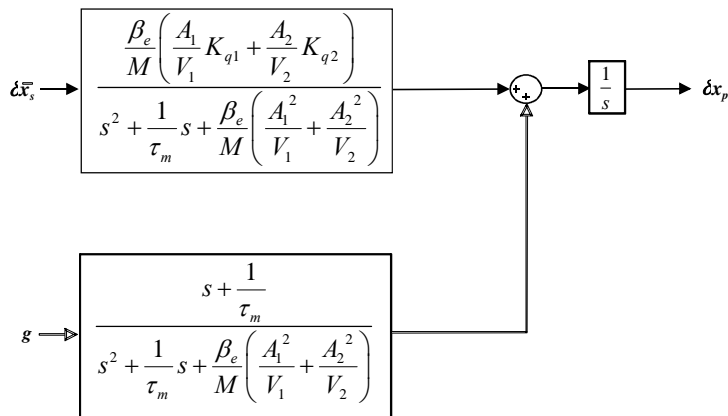


Figura 6.16 Diagrama de blocos simplificado tendo como entradas o deslocamento da gaveta da válvula e a aceleração da gravidade

Para o cilindro a actuar na horizontal, a função de transferência vem dada por:

$$\frac{dx_p}{d\bar{x}_s} = \frac{K \times w_n^2}{s(s^2 + 2 \times x \times w_n + w_n^2)} \quad (6.12)$$

$$\text{com } \begin{cases} K = \frac{V_1 V_2 f}{t_m (A_1 V_2 K_{q1} + A_2 V_1 K_{q2})} + \frac{A_1 V_2 K_{q1} + A_2 V_1 K_{q2}}{A_1^2 V_2 + A_2^2 V_1} \\ w_n^2 = \frac{f}{M \times t_m} + \frac{b_e \frac{A_1^2}{V_1} + \frac{A_2^2}{V_2} \frac{\ddot{\theta}}{\theta}}{M} \\ x = \frac{1}{t_m} + \frac{f}{M} \\ 2w_n \end{cases} \quad (6.13)$$

As características estáticas da válvula, para o ponto de linearização, têm os seguintes valores medidos:

$$k_{q1} = \frac{\partial Q_1}{\partial \bar{x}_s} = 4.76 \times 10^{-4} \sqrt{\frac{P_s}{2P_n}} \text{ m}^3 \text{ s}^{-1}; \quad k_{q2} = \frac{\partial Q_2}{\partial \bar{x}_s} = 4.76 \times 10^{-4} \sqrt{\frac{P_s}{2P_n}} \text{ m}^3 \text{ s}^{-1}$$

$$k_{p1} = \frac{\partial P_1}{\partial \bar{x}_s} = 19P_s \text{ Pa}; \quad k_{p2} = \frac{\partial P_2}{\partial \bar{x}_s} = -15.6P_s \text{ Pa}$$

Outros parâmetros usados foram:  $V_{01} = 3 \times 10^{-5} \text{ m}^3$ ,  $V_{02} = 5 \times 10^{-5} \text{ m}^3$ ,  $A_1 = 1.2566 \times 10^{-3} \text{ m}^2$  e  $A_2 = 8.7650 \times 10^{-4} \text{ m}^2$ .

Os diagramas de Bode para o modelo em espaço de estados e para a função de transferência, para um movimento na horizontal,  $M = 80 \text{ Kg}$  e  $X_{p0} = 82 \text{ mm}$ , são apresentados na Figura 6.17. Nota-se que a simplificação realizada através da uniformização das constantes de tempo tem pouca influência na resposta em frequência do modelo linear.

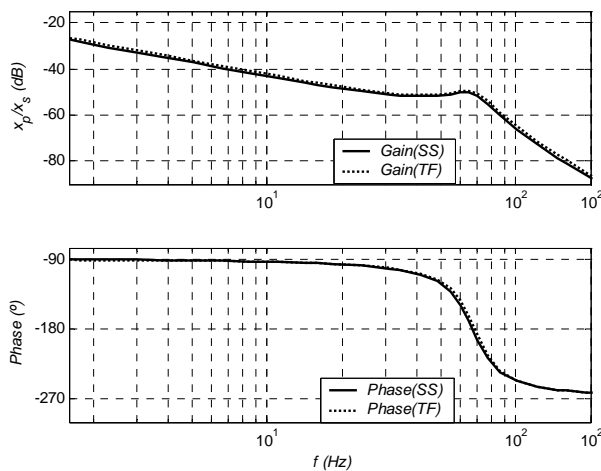


Figura 6.17 Diagrama de Bode do espaço de estados (6.9) e da função de transferência (6.12)

### 6.2.1.2 Experiências para determinação da frequência natural amortecida

As experiências para a determinação da frequência natural amortecida foram realizadas de acordo com a Figura 6.18. Foi aplicada uma onda sinusoidal de frequência variável, com passos de 1 Hz (durante 1 segundo) até 130 Hz, e com amplitude igual a 10% do valor máximo permitido na carta de controlo da válvula. Como mostra o esquema da Figura 6.18, o sistema (real e simulado) foi operado em malha fechada com um ganho proporcional pequeno, de modo a manter o êmbolo em torno da posição de equilíbrio  $X_{p0}$ . Foram realizadas várias experiências com o cilindro hidráulico para medição da frequência natural amortecida para diferentes posições equilíbrio, diferentes pressões de trabalho e diferentes massas. Exemplos dos sinais de aceleração e posição do êmbolo são apresentados na Figura 6.19 e na Figura 6.20 para posições diferentes de equilíbrio. Para a medição da aceleração real foi instalado um acelerómetro de vibrações (com uma gama de  $\pm 80g$ ).

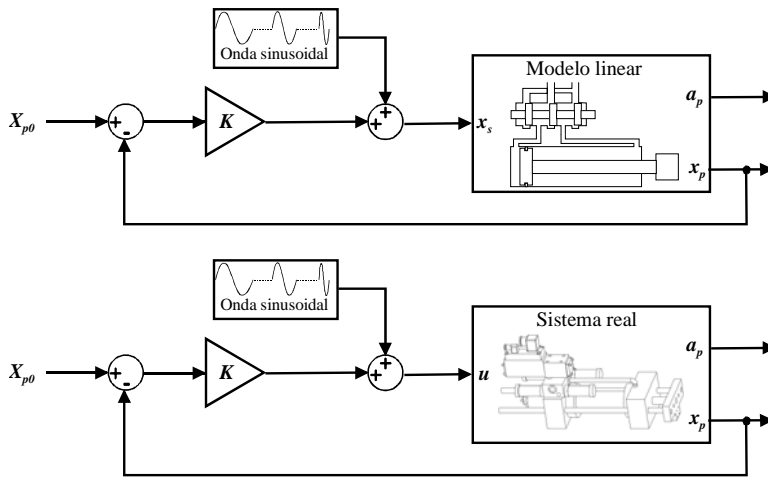


Figura 6.18 Diagrama de blocos para medição da frequência natural amortecida

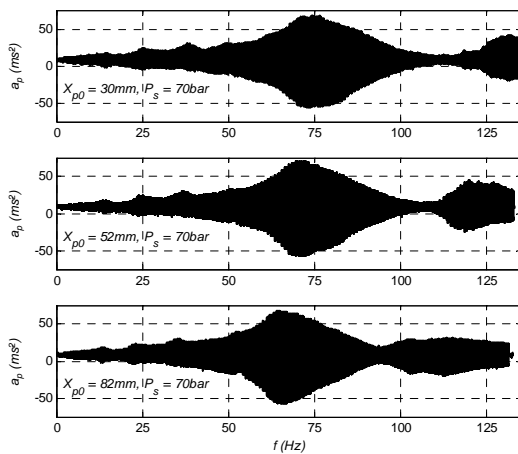


Figura 6.19 Sinais de aceleração do êmbolo medidos para diferentes posições de equilíbrio,  $X_{p0}$



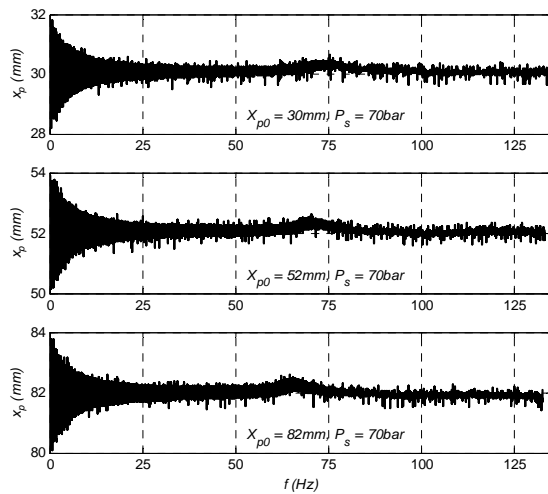


Figura 6.20 Sinais de posição do êmbolo medidos para diferentes posições de equilíbrio,  $X_{p0}$

Com um algoritmo de determinação da frequência de ocorrência da amplitude máxima para o sinal de aceleração pode obter-se a frequência natural amortecida do sistema,  $f_d$ . A Figura 6.21 mostra os valores medidos para a frequência natural não amortecida em função da posição do êmbolo.

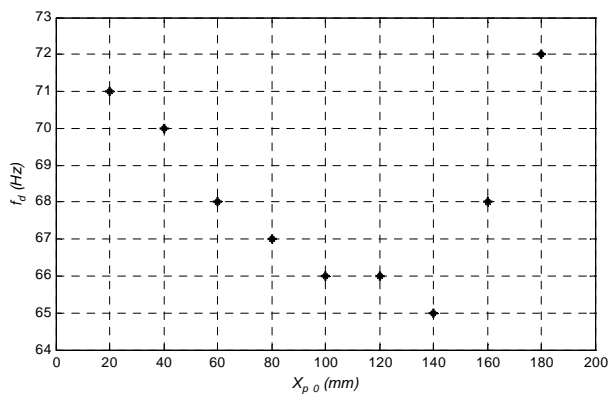


Figura 6.21 Valores medidos para a frequência natural em função da posição do êmbolo

### 6.2.1.3 Estimação do módulo de compressibilidade efectivo

O módulo de compressibilidade efectivo,  $\beta_e$ , e o coeficiente de atrito viscoso,  $f$ , foram estimados através de um processo de optimização; este processo procura minimizar a distância, no plano aceleração – frequência, entre as acelerações máximas dos êmbolos, uma obtida do sistema real, e a outra da simulação do espaço de estados (6.9). O sinal de aceleração medido foi processado e filtrado de modo a retirar a envolvente do sinal. O modelo linear (6.9) foi simulado para as mesmas frequências, posições e amplitudes da válvula. Os valores da aceleração do modelo simulado foram

igualmente processados, sendo a sua envolvente retirada. As figuras 6.22 e 6.23 mostram os resultados desta experiência para diferentes pressões na fonte,  $P_s$ , e para movimentos do êmbolo em torno da posição de  $82\text{mm}$  (correspondente a volumes iguais nas duas câmaras do cilindro).

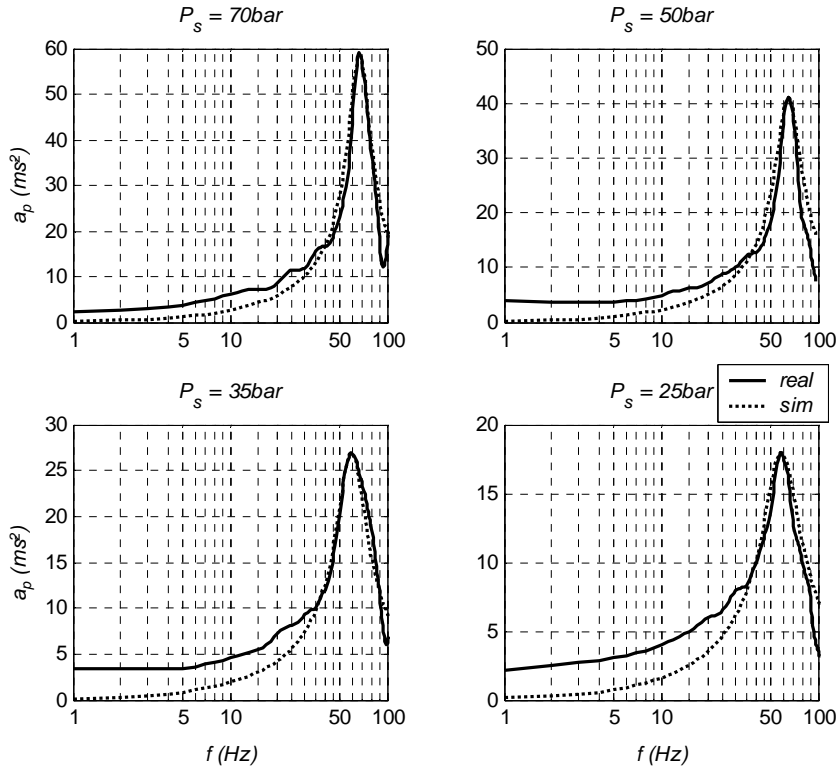


Figura 6.22 Envolvente da aceleração real obtida através do acelerómetro e da aceleração obtida pela simulação do modelo (6.9) nas mesmas condições, para diferentes pressões da fonte

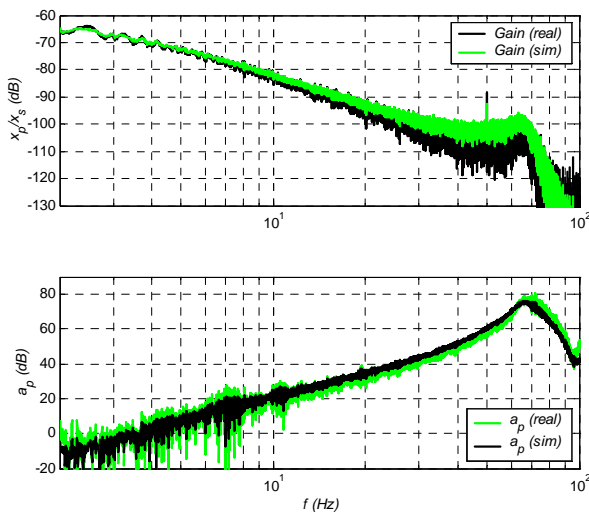


Figura 6.23 Espectro de potência dos ganhos  $(x_p/\bar{x}_s)$  e das acelerações  $a_p$  (reais e simulados)

Na Figura 6.23 mostra-se o espectro de potência dos ganhos  $(x_p/\bar{x}_s)$  e das acelerações  $a_p$  (reais e simulados), obtido através da aplicação da FFT (*Fast Fourier Transform*) em pós processamento aos respectivos sinais (ver exemplos desses sinais na Figura 6.19).

Os parâmetros (módulo de compressibilidade efectiva nas duas câmaras,  $b_{e1}$  e  $b_{e2}$ , e o coeficiente de atrito viscoso,  $f$ ) obtiveram-se através de optimização, sendo a função de custo representada pela proximidade dos pontos de máxima aceleração no modelo simulado e na experiência real. O modelo em espaço de estados (6.9) foi simulado com o vector de estado inicial  $s_0 = \begin{bmatrix} \dot{P}_{10} & P_{20} & V_{p0} & X_{p0} \end{bmatrix}$ .  $P_{10}$  e  $P_{20}$  foram obtidos a partir de (6.1) e (6.2), respectivamente. Os resultados da simulação, apresentados na Figura 6.22, foram obtidos para os seguintes vectores de estado iniciais e parâmetros:

$$s_0 = \begin{bmatrix} 25 & 38 & 0 & 0.082 \end{bmatrix} \cdot 10^5, P_s = 70bar, b_{e1} = 7700bar, b_{e2} = 9600bar, f = 10200Nsm^{-1}$$

$$s_0 = \begin{bmatrix} 17 & 26 & 0 & 0.082 \end{bmatrix} \cdot 10^5, P_s = 50bar, b_{e1} = 7270bar, b_{e2} = 8970bar, f = 11100Nsm^{-1}$$

$$s_0 = \begin{bmatrix} 11 & 17 & 0 & 0.082 \end{bmatrix} \cdot 10^5, P_s = 35bar, b_{e1} = 6180bar, b_{e2} = 7300bar, f = 11600Nsm^{-1}$$

$$s_0 = \begin{bmatrix} 7 & 11 & 0 & 0.082 \end{bmatrix} \cdot 10^5, P_s = 25bar, b_{e1} = 5950bar, b_{e2} = 6920bar, f = 13800Nsm^{-1}$$

As frequências naturais amortecidas e os valores máximos para as acelerações do êmbolo, em função da pressão de trabalho, são mostradas na Figura 6.24.

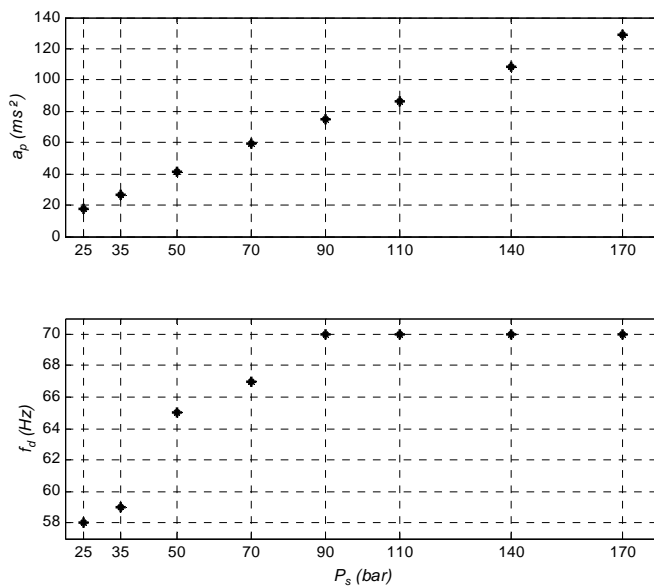


Figura 6.24 Frequências naturais e acelerações máximas, em função da pressão da fonte, obtidas entre 25bar e 170bar

A Figura 6.25 mostra o módulo de compressibilidade efectivo médio  $b_e = (b_{e1} + b_{e2})/2$ , em função da pressão média nas câmaras do cilindro  $P = (P_1 + P_2)/2$ .

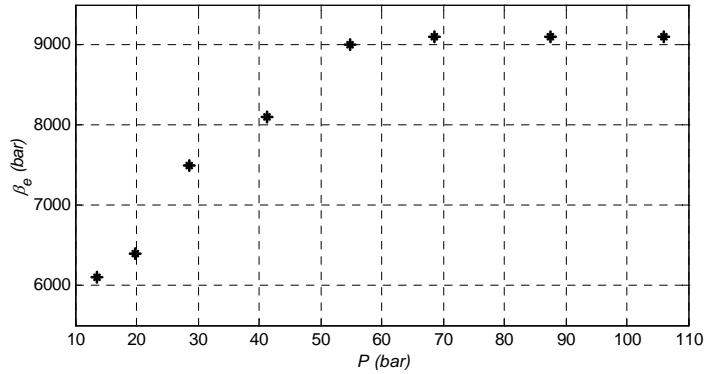


Figura 6.25 Módulo de compressibilidade efectivo médio, em função da pressão média nas câmaras do cilindro

### 6.2.1.4 Parametrização dos modelos para o módulo de compressibilidade efectivo

A parametrização dos modelos para o módulo de compressibilidade efectivo foi realizada através de optimização. O resultado da optimização do modelo de Jinghong [Jinghong94] com três (6.14) e dois (6.15) parâmetros é apresentado na Figura 6.26.

$$b_e = \frac{10^5 + P}{A \times P^2 + B \times P + C} \quad (6.14)$$

$$b_e = \frac{10^5 (1 + P \times 10^{-5})}{B \times P + C} \quad (6.15)$$

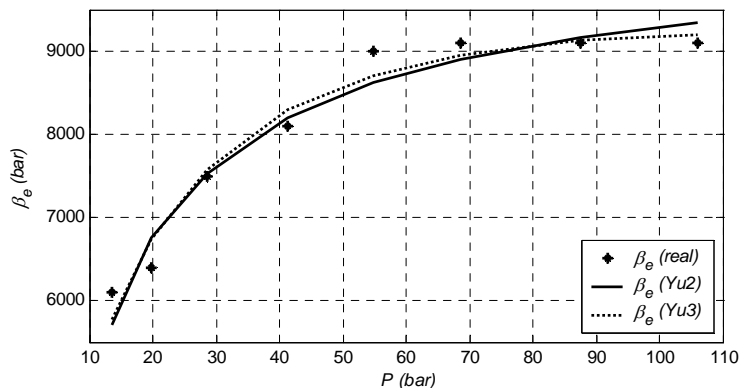


Figura 6.26 Módulo de compressibilidade efectivo medido e simulado pelos modelos: modelo 2 e modelo 3 (ver secção 4.1.2).

Os parâmetros estimados para (6.14) são  $A = 8.33 \cdot 10^{-18}$ ,  $B = 8.86 \cdot 10^{-10}$  e  $C = 1.30 \cdot 10^{-3}$ , enquanto que para (6.15) são  $B = 9.71 \cdot 10^{-10}$  e  $C = 1.15 \cdot 10^{-3}$ .

### 6.2.2 Medição da condutância de fugas internas do cilindro

As fugas internas do cilindro são consideradas laminares e representadas por uma condutância de fugas:

$$g_{lkc} = \frac{q_{lkc}}{(P_1 - P_2)} \quad (6.16)$$

O processo de medição da condutância de fugas internas do cilindro foi o seguinte: na posição  $x_p=0$ , o terminal 2 da válvula foi bloqueado, o terminal 1 foi aberto para a atmosfera e o cilindro foi libertado com uma carga elevada na posição vertical (ver Figura 6.12); a posição do êmbolo e a pressão na câmara 2 do cilindro foram monitorizadas durante um longo período de tempo ( $\Delta t$ ), de modo a obter uma velocidade constante em regime estacionário,  $v_{ps}$ . A condutância de fugas pode então ser calculada por:

$$g_{lkc} = \frac{v_{ps} \times A_2}{P_2} \quad (6.17)$$

A posição do êmbolo e a pressão na câmara 2 do cilindro foram monitorizadas durante um período de tempo  $Dt = 200\text{seg}$ . O deslocamento total, no intervalo de tempo, foi de  $1504\text{mm}$ , e o valor médio da pressão foi  $P_2 = 5.28\text{bar}$ .

O valor obtido para a condutância interna de fugas foi  $g_{lkc} = 1.248 \times 10^{-14} \text{m}^3 \text{s}^{-1} \text{Pa}^{-1}$ .

### 6.2.3 Parametrização do modelo de atrito de LuGre para o cilindro

A forma de estimar os parâmetros estáticos ( $f_{co}, f_s, v_s, k_v$ ) e dinâmicos ( $s_0, s_1$ ) do modelo de LuGre é apresentada em seguida. Recordando a secção 3.4.2, o modelo de LuGre é definido pelas seguintes equações:

$$\frac{dz}{dt} = v_p - \frac{|v_p|}{g(v_p)} z \quad (6.18)$$

$$g(v_p) = \frac{1}{s_0} f_{co} + f_s \times e^{-\left(\frac{v_p}{v_s}\right)^2} \quad (6.19)$$

$$f_a = s_0 z + s_1 \frac{dz}{dt} + k_v v_p \quad (6.20)$$

### 6.2.3.1 Identificação dos parâmetros estáticos

Os parâmetros estáticos são estimados através da curva velocidade versus força de atrito. A força de atrito foi medida em regime de velocidade constante, sendo a velocidade calculada por diferenciação do sinal de posição, obtido através de uma régua óptica com uma resolução de  $1mm$ . As pressões nos terminais da válvula e nas câmaras do cilindro foram monitoradas através de quatro transdutores de pressão ( $P_1, P_2, P_s, P_t$ ). O tempo de amostragem utilizado foi de  $5ms$ , e as velocidades e forças foram calculadas usando 20 amostras para minimizar os efeitos do ruído. As experiências a velocidade constante foram realizadas com controlo da velocidade em malha fechada. A força de atrito foi medida para velocidades constantes entre  $-0.2ms^{-1}$  e  $0.2ms^{-1}$ .

A plataforma para parametrização do modelo de atrito é a representada na Figura 6.11, mas com funcionamento na horizontal.

A velocidade constante (uma situação em regime permanente,  $a_p = 0$ ) e com a plataforma na horizontal, a força de atrito pode ser calculada através dos valores das pressões nas câmaras pois a carga aplicada ao cilindro é nula:

$$F_a = P_1 A_1 - P_2 A_2 \quad (6.21)$$

A velocidade constante, a variável de estado do modelo de LuGre, equações (6.18), (6.19) e (6.20) é constante,  $dz/dt = 0$ , e a força de atrito pode ser estimada da seguinte forma

$$\hat{f}_a = \hat{f}_{co} + \hat{f}_s e^{-\left(\frac{v_p}{\hat{v}_s}\right)^2} \frac{\partial}{\partial t} \text{sign}(v_p) + \hat{k}_v v_p \quad (6.22)$$

onde  $(\hat{f}_{co}, \hat{f}_s, \hat{v}_s, \hat{k}_v)$  são os parâmetros estáticos estimados e  $\hat{f}_a$  é a força de atrito estimada.

Para a estimação dos parâmetros foi usado o método dos mínimos quadrados, com a função de custo

$$cf = \sum_{i=1}^n (f_a(v_i) - \hat{f}_a(v_i))^2 \quad (6.23)$$

onde  $v_i$  são as velocidades medidas.

A função de custo foi calculada para cada conjunto de parâmetros  $(\hat{f}_{co}, \hat{f}_s, \hat{v}_s, \hat{k}_v)$  através do algoritmo *simplex* [Lagarias98] implementado na função *fminsearch* da plataforma Matlab®. Os

valores iniciais para os parâmetros foram retirados directamente da curva velocidade versus força de atrito medida. Os parâmetros estáticos estimados são aqueles que minimizam a função de custo. Para um modelo simétrico, isto é, tendo os mesmos parâmetros para velocidades negativas e positivas, o conjunto de parâmetros estimados é:

$$\hat{f}_{co} = 101.784N, \hat{f}_s = 153.049N, \hat{v}_s = 0.019ms^{-1}, \hat{k}_v = 1089.735Nsm^{-1}$$

A análise das curvas reais da velocidade versus força de atrito denota diferentes comportamentos para velocidades simétricas. O modelo pode então ser melhorado, considerando parâmetros diferentes para velocidades negativas e positivas. Os novos parâmetros estimados são:

$$\hat{f}_{con} = 89.258N, \hat{f}_{sn} = 160.281N, \hat{v}_{sn} = 0.0251ms^{-1}, \hat{k}_{vn} = 1387.490Nsm^{-1}$$

$$\hat{f}_{cop} = 110.179N, \hat{f}_{sp} = 150.767N, \hat{v}_{sp} = 0.0152ms^{-1}, \hat{k}_{vp} = 818.445Nsm^{-1}$$

onde os índices  $n$  e  $p$  significam velocidades negativas e positivas. Os resultados são apresentados na Figura 6.27.

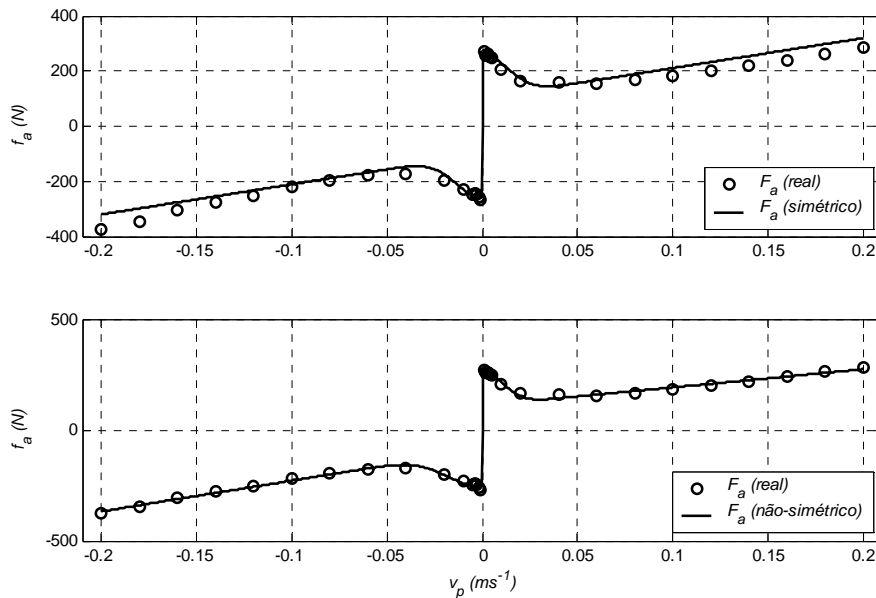


Figura 6.27 Velocidade versus força de atrito em regime permanente

### 6.2.3.2 Identificação dos parâmetros dinâmicos

Em [Canudas97] e [Lischinsky99], no âmbito da modelação de um motor DC, são apresentados dois métodos para a identificação dos parâmetros dinâmicos do modelo de LuGre. Ambos os métodos se baseiam na medida precisa dos binários disponíveis dos actuadores.

Por exemplo, utilizando o método proposto por Lischinsky, em [Lischinsky99], é necessário aplicar uma força com uma variação lenta e inferior à força de *breakaway*, de tal forma que o sistema não abandone o estado de colagem. Neste caso, o deslocamento medido coincide com a variável de estado do modelo de LuGre ( $z$ ). Nessa situação, a aceleração e a velocidade são praticamente nulas e a rigidez,  $s_0$ , pode ser calculada a partir do micro-deslocamento produzido por uma variação da força disponível, isto é,  $s_0 = DF/Dx_p$ . Muito embora o micro-deslocamento possa ser medido, o mesmo não acontece para a força hidráulica que não pode ser medida com a precisão necessária para usar os métodos propostos em [Canudas97] e [Lischinsky99].

O método alternativo proposto consiste em fazer coincidir a força hidráulica medida com a produzida por um modelo não linear do conjunto válvula + cilindro, nas mesmas condições. O modelo de válvula usado foi o modelo estático 11, apresentado na secção 5.1.3, e os parâmetros estáticos para o modelo de atrito foram os calculados na secção anterior para o modelo não simétrico. Foram realizadas experiências em malha aberta, de modo a permitir maior sensibilidade aos efeitos do atrito. O sistema a trabalhar na horizontal foi excitado com uma onda sinusoidal aplicada à válvula, com amplitude suficiente para que o sistema entre e saia do estado de colagem.

Foi usado um método de otimização idêntico ao usado na estimação dos parâmetros estáticos com a função de custo seguinte:

$$cf(F_h, F_{hm}, \hat{s}) = \sum_{k=1}^n (F_h(k) - F_{hm}(k, \hat{s}))^2 \quad (6.24)$$

onde  $F_h(k)$  é a amostra  $k$  da força hidráulica real (tempo de amostragem igual a 10ms) e  $F_{hm}(k, \hat{s})$  é a força hidráulica, no mesmo instante temporal, que resulta da simulação do modelo não linear do conjunto válvula + cilindro, com as mesmas condições iniciais. A utilização da força hidráulica como comparação resulta da seguinte aproximação.

$$M \frac{dv_p}{dt} = F_h - F_a \quad (6.25)$$

Para a massa usada (êmbolo + haste), os valores máximos de  $M \frac{dv_p}{dt}$  estão abaixo de 0.05N, que é um valor desprezável quando comparado com a força hidráulica e com a força de atrito.

Os sinais provenientes dos sensores de pressão foram filtrados antes do cálculo da força hidráulica real ( $F_a = P_1A_1 - P_2A_2$ ). Com base na função de custo (6.24) foram estimados os seguintes valores para os parâmetros dinâmicos do modelo de LuGre:  $s_0 = 2.114 \cdot 10^7 Nm^{-1}$  e



$s_1 = 2.914 \cdot 10^3 Nsm^{-1}$ . As figuras 6.28, 6.29 e 6.30 mostram a comparação entre as pressões nas câmaras do cilindro, as forças hidráulicas e as posições e velocidades, quando na simulação foram usados os parâmetros não simétricos para os parâmetros estáticos do modelo de atrito de LuGre.

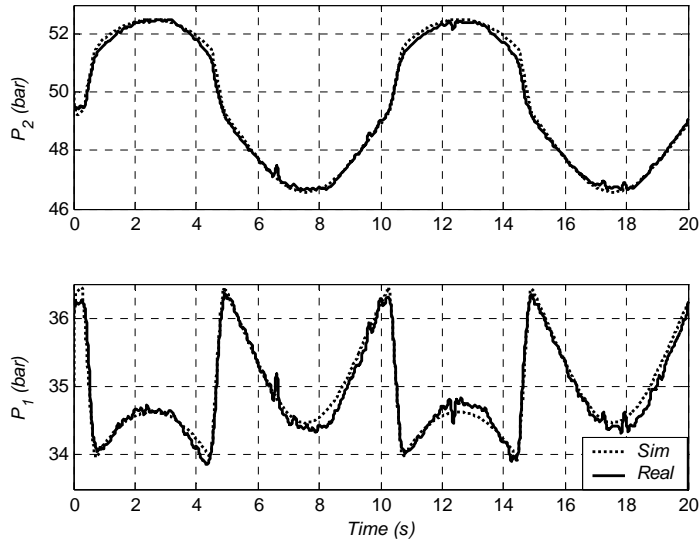


Figura 6.28 Pressões reais e simuladas nas câmaras do cilindro ( $s_0 = 2.114 \cdot 10^7 Nm^{-1}$  e  $s_1 = 2.914 \cdot 10^3 Nsm^{-1}$ )

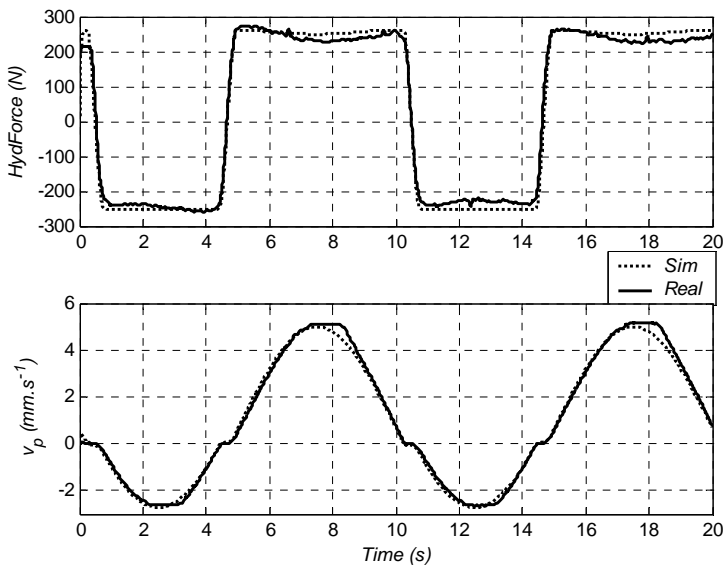


Figura 6.29 Velocidades e forças hidráulicas reais e simuladas ( $s_0 = 2.114 \cdot 10^7 Nm^{-1}$  e  $s_1 = 2.914 \cdot 10^3 Nsm^{-1}$ )

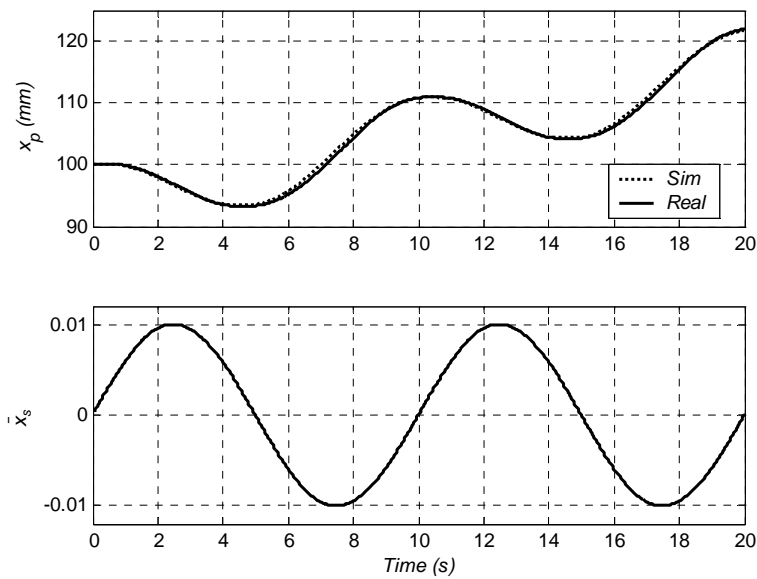


Figura 6.30 Sinal de entrada na válvula e posição simulada e real do êmbolo ( $s_0 = 2.114 \cdot 10^7 Nm^{-1}$  e  $s_1 = 2.914 \cdot 10^3 Nsm^{-1}$ )

Nesta secção foram identificados os parâmetros, de alguns dos modelos apresentados neste trabalho, com vista à sua utilização em experiências de simulação apresentadas nas secções seguintes.

Foram parameterizados os modelos necessários à simulação de um servomecanismo hidráulico, ilustrado na Figura 6.11 e na Figura 6.12. A identificação do módulo de compressibilidade efectivo foi realizada através da comparação de resultados simulados e experimentais; a aceleração produzida através da simulação de um modelo linear do servomecanismo hidráulico foi aproximada da medida em experiências reais, quando os dois sistemas foram excitados com um mesmo sinal sinusoidal de frequência variável. Outro dos modelos parameterizados foi o modelo de atrito de LuGre; a metodologia proposta para a determinação dos parâmetros dinâmicos consistiu em fazer coincidir a força hidráulica medida experimentalmente, em experiências a baixa velocidade e com inversão do seu sentido, com a força hidráulica simulada através de um modelo não linear do conjunto válvula + cilindro a operar nas mesmas condições. Pelos resultados obtidos pode concluir-se da adequabilidade dos métodos propostos para as diferentes parametrizações.

### 6.3 *Hardware-in-the-loop*

*Hardware-in-the-loop* (HILS) refere-se a uma simulação onde alguns dos componentes são reais e não simulados. É considerada a técnica mais segura e de mais baixo custo para teste de componentes reais em ambientes virtuais. A maioria dos componentes reais são substituídos por modelos matemáticos e os componentes a testar são inseridos na malha fechada. Uma das razões para inserir componentes numa simulação surge, muitas vezes, pela inexistência de um conhecimento cabal das suas características, ou as suas características são muito complexas, ou então quando existe a necessidade de teste dos próprios componentes reais, como é o caso do teste de controladores. Outra das razões advém do próprio processo de projecto de novos sistemas de controlo que cumpram determinadas especificações. Pode-se, por exemplo, partir de uma simulação pura em tempo real, com os modelos de todos os componentes do sistema, de modo a cumprir determinadas especificações; à medida que os componentes vão sendo fabricados, vão substituindo os respectivos modelos matemáticos na simulação, de tal forma que as características dos outros componentes poderão ser re-ajustadas de modo a cumprir as especificações iniciais; este processo permite, por exemplo, reduzir o número de iterações na maquinação e fabrico de peças para os diferentes componentes do sistema.

A HILS é, também, uma ferramenta bastante útil para avaliação e desenvolvimento de controladores, oferecendo um risco nulo na experimentação de diferentes técnicas e metodologias de controlo sem necessidade da plataforma real para teste. Desta forma, é possível poupar investimentos e evitar consequências perigosas resultantes de erros no projecto inicial dos controladores, permitindo, assim, a identificação e eliminação desses erros. O uso da HILS para as diferentes regiões de operação, incluindo modos com falhas, permite a selecção das estratégias adequadas de controlo na plataforma real, pois facilita a repetição de testes de desempenho.

A Figura 6.31 mostra as diferentes possibilidades de interacção de componentes reais com versões simuladas de outros componentes num sistema de controlo típico. Como se pode facilmente depreender, existem alguns caminhos na malha que não são possíveis; por exemplo, não é possível um processo real ser monitorado através de sensores simulados ou actuadores simulados actuarem em processos reais. A actuação, através de actuadores reais, em processos simulados tem utilidade, nomeadamente nas denominadas técnicas de *load simulation*, que consistem na utilização de um processo simulado para avaliação de desempenho de actuadores reais; considere-se, por exemplo, a avaliação do desempenho de uma válvula hidráulica, num qualquer actuador hidráulico simulado, em que as condições de carga da válvula se fazem sentir através de válvulas de interface actuadas a partir da simulação [Jansson94]. A monitorização de processos simulados através de sensores reais, embora com aplicações não tão evidentes, pode ser utilizada quando se quer avaliar o desempenho

de um determinado sensor; neste caso é necessária uma interface adequada, que pode envolver um actuador real para converter os resultados da simulação em quantidades mensuráveis pelo sensor.

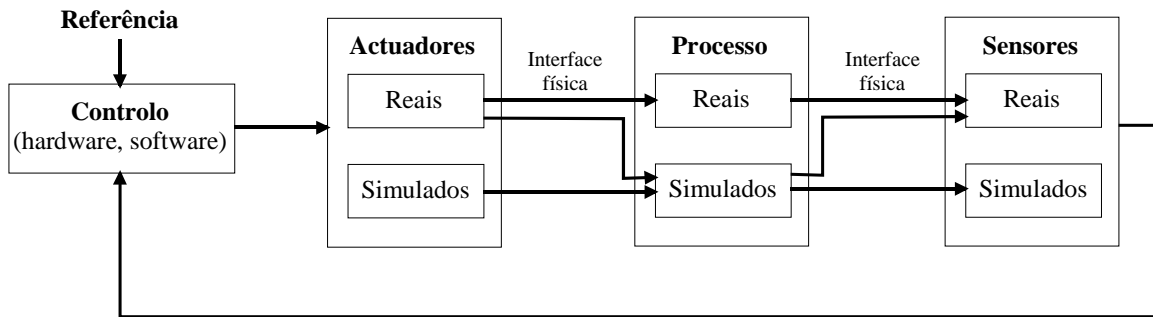


Figura 6.31 Possibilidades de simulação com *hardware-in-the-loop* em sistemas de controlo

Uma condição necessária para a realização de experiências de HILS é a capacidade de simulação em tempo real dos modelos matemáticos dos componentes.

### Simulação em tempo real

A simulação em tempo real (STR) pode definir-se como aquela em que as variáveis de um sistema simulado representam fielmente não só os valores do sistema real, mas também os instantes temporais associados a esses valores. A simulação em tempo real envolve problemas diferentes das outras simulações no que diz respeito aos métodos numéricos a utilizar. Na simulação em tempo não real, o objectivo principal dos métodos numéricos é a diminuição do tempo de simulação, para uma dada precisão dos valores obtidos para as variáveis. Estes métodos recorrem a sofisticados mecanismos que ajustam o passo de integração e/ou controlam a ordem do método para manter a estabilidade numérica e a obtenção da precisão pretendida. Na simulação em tempo real com HIL, o passo de integração é fixo pois, devido às contingências da simulação, o hardware de simulação pode ter de comunicar com periféricos de hardware durante a simulação. A comunicação entre a simulação e os periféricos ocorre com um determinado tempo de amostragem e a simulação tem de fornecer os resultados dentro desse intervalo de amostragem. Desta forma, o principal objectivo da simulação em tempo real é garantir que o tempo de cálculo, para obtenção dos valores para as variáveis, é inferior ao passo de integração.

#### 6.3.1 Técnicas e métodos numéricos na STR

Na simulação em tempo real são usados métodos de integração de passo fixo que podem ser explícitos ou implícitos (ver secção 2.2.2). Os métodos de integração de passo fixo explícitos têm

particulares dificuldades com os sistemas rígidos (sistemas em que a relação entre os valores próprios, correspondentes às dinâmicas rápidas e às dinâmicas lentas, é grande), pois o passo de integração é limitado por problemas de estabilidade numérica. Por exemplo, quando se utiliza o método explícito de Euler, o passo de integração tem que ser inferior à menor constante de tempo do sistema, para garantir a estabilidade do método [Heath97]; se o passo for demasiado grande a trajectória começa a oscilar e diverge. A solução para estes comportamentos instáveis passa, normalmente, pela utilização de métodos implícitos. Contudo, o uso destes métodos implica a resolução, em cada passo, de grandes sistemas de equações não lineares cuja dimensão é pelo menos igual ao número de variáveis de estado do sistema. Para a resolução destes sistemas de equações não lineares são usados métodos numéricos iterativos. O processo iterativo termina quando a precisão desejada é atingida e, portanto, não é possível prever o número de iterações necessário em cada passo. Além disso, nos sistemas de equações de estrutura variável (resultantes, por exemplo, da modelação de sistemas híbridos), os sistemas de equações a resolver podem, ainda, ser diferentes em diferentes regiões de operação do sistema modelado e, portanto, não se pode tirar partido da estrutura do modelo na simulação em tempo real [Scheila02]. Os métodos implícitos resolvem o problema da estabilidade numérica e permitem a utilização de passos de integração maiores, sendo a restrição ao passo de integração definida através da precisão que se pretende obter.

Em aplicações industriais os sistemas a simular são, muitas vezes, modelados por grandes sistemas de DAEs rígidas. Na maioria dos casos a rigidez é devida a componentes com dinâmicas rápidas (por exemplo componentes hidráulicos, eléctricos, controladores, ...) quando comparadas com as escalas temporais de interesse (por exemplo o movimento de peças mecânicas).

Como se depreende do exposto, os métodos explícitos e os métodos implícitos têm vantagens e desvantagens quando usados na simulação em tempo real. Nos últimos anos foram apresentadas alternativas que exploram as vantagens de ambos os métodos para permitir a simulação em tempo real de sistemas mais complexos, nomeadamente os sistemas multi-domínio que resultam, normalmente, em sistemas de equações rígidas.

Uma das técnicas, chamada *multirate integration*, consiste em utilizar a discretização explícita com passos de integração diferentes, ajustados às dinâmicas lentas e às dinâmicas rápidas. A principal vantagem resulta da inexistência de métodos iterativos. Esta técnica pode ser usada em sistemas onde as equações diferenciais possam ser divididas em subsistemas, pouco acoplados ou desacoplados, cujas dinâmicas tenham diferentes escalas temporais [Buzdugan99]. Nos subsistemas de dinâmica lenta é usado um menor número de passos de integração, do que nos subsistemas de dinâmicas rápidas.

Outro método proposto consiste na linearização, através da diferenciação simbólica, do sistema de equações, como forma de evitar o processo iterativo resultante da utilização de métodos implícitos [Gawthrop00]. Uma das vantagens referidas pelo autor é que a diferenciação simbólica tem de ser realizada apenas antes do início da simulação, sendo, além disso, uma diferenciação “exacta”.

O método *mixed-mode integration*, [Schiela00], [Elmqvist02], permite a redução da dimensão do sistema de equações não lineares resultantes da discretização implícita. Esta técnica usa a discretização explícita para os estados de dinâmica lenta e a discretização implícita para os estados com dinâmica rápida. Schiela, em [Schiela00], propõe a linearização e análise dos valores próprios, para a partição do vector de estado em estados de dinâmica lenta e estados com dinâmica rápida. A discretização explícita da “parte” lenta do sistema de equações quebra o acoplamento entre os componentes implícitos no sistema de equações, podendo os sistemas desacoplados ser tratados de forma mais eficiente através de manipulação simbólica.

Outro método, chamado *inline integration*, consiste em inserir as fórmulas de discretização (explícita ou implícita) nas equações diferenciais, aquando da compilação do modelo. Através de manipulações simbólicas, as equações diferenciais contínuas do modelo são substituídas pelas correspondentes equações discretas no tempo. Se a forma de discretização for implícita, Elmqvist, em [Elmqvist94] e [Elmqvist95], propõe um conjunto de transformações que reduzem o número de equações simultâneas para pequenos blocos de equações, por forma a que possam ser resolvidas mais eficientemente: de forma simbólica, na altura da compilação, ou numericamente, quando em execução.

Estes dois últimos métodos estão disponíveis, isoladamente ou em conjunto, na plataforma Dymola, tendo sido alguns resultados publicados em [Soejima02] e [Elmqvist02].

A partir de uma determinada complexidade, os modelos não podem ser simulados em tempo real a baixo custo, sendo necessário aumentar o desempenho do hardware de simulação e/ou diminuir a complexidade dos modelos através da utilização de, por exemplo, modelos semi-empíricos e modelos híbridos, abordados em capítulos anteriores. Por exemplo, Gonthier, em [Gonthier98], propõe modelos reduzidos de actuadores hidráulicos, para simulação em tempo real, sem redução significativa da precisão; Münzer, em [Münzer01], propõe um modelo simplificado para uma grua móvel, de modo a permitir a simulação com visualização em tempo real. A diminuição do passo de integração, para um mesmo método de integração, implica a redução da complexidade dos modelos, de modo a que se obtenham os novos valores para as variáveis do modelo dentro do intervalo de tempo definido pelo passo de integração. Quanto ao aumento do desempenho, este tem a sua máxima expressão nos sistemas de processamento distribuído, que permitem a simulação em tempo real de sistemas bastante complexos [Engelbrecht99], [Lane01], [Stolpe98].

### 6.3.2 Aplicações e plataformas para HILS

Aplicações da simulação em tempo real encontram-se nas áreas do controlo, nomeadamente no controlo preditivo baseado em modelos, em simuladores interactivos (simuladores de voo ou máquinas de diversão) e em experiências de simulação com *hardware-in-the-loop*. De entre as plataformas de hardware, para simulação em tempo real de mais baixo custo, destacam-se as plataformas baseadas em computadores pessoais, equipados com sistemas operativos de tempo real, e plataformas baseadas em processadores digitais de sinal (DSPs).

As plataformas para STR fornecem, na sua generalidade, software e/ou hardware que permite a simulação em tempo real de modelos de uma forma automatizada, isto é, sem necessidade de desenvolvimento de qualquer código. De seguida enunciam-se algumas dessas plataformas.

A plataforma VisSim ([www.vissim.com](http://www.vissim.com) 2003), com a opção *Real-Time*, permite simular sistemas e controladores em tempo real com possibilidade de interface com periféricos reais. Esta plataforma permite a simulação em tempo real e a aquisição de dados em ambiente Windows.

O RT-LAB ([www.opal-rt.com](http://www.opal-rt.com) 2003) é um pacote de software que fornece ferramentas para correr e monitorar simulações em diversas plataformas. Uma das particularidades consiste na possibilidade de execução de modelos de Simulink ou SystemBuild em plataformas de processamento distribuído. Inclui também uma grande variedade de *drivers* para placas de I/O.

A Humusoft ([www.humusoft.cz](http://www.humusoft.cz) 2003) oferece uma ferramenta denominada *Extended Real Time Toolbox*, em conjunto com uma série de *drivers* para placas de I/O, que permite a execução em tempo real de modelos em Simulink, em ambiente Windows ou em plataformas com sistemas operativos de tempo real.

O RealSim ([www.windriver.com](http://www.windriver.com) 2003) fornece um ambiente gráfico completo para simulação e para *hardware-in-the-loop* em plataformas PC.

A Mathworks ([www.mathworks.com](http://www.mathworks.com) 2003) disponibiliza duas soluções de baixo custo para simulação em tempo real ou HILS. Permite a execução de modelos Simulink em, por exemplo, PC dedicado (xPC) ou em núcleo tempo real através do *Real Time Windows Target* (RTWT).

A dSPACE ([www.dspace.com](http://www.dspace.com) 2003) apresenta um conjunto de soluções de software e hardware com diferentes desempenhos para a realização de experiências HILS, desde simples cartas baseadas em DSPs a grandes sistemas com processamento distribuído, para simulação em tempo real de sistemas complexos. Oferece ferramentas de software para comando e monitorização de simulações tempo real e ferramentas de programação do hardware, de forma directa ou em interface com a plataforma Matlab/Simulink.

### 6.3.3 Exemplos de aplicações de HILS

O desenvolvimento de unidades de controlo electrónicas, conhecidas como ECUs [Hanselmann96], em especial para a indústria automóvel, vieram dinamizar a utilização das técnicas de HILS (anteriormente com uso quase exclusivo da indústria aeronáutica e aeroespacial) em variadas áreas da engenharia. A HILS permite testar as ECUs num ambiente de simulação em malha fechada. Os motores, veículos e outros componentes que os ECUs controlam são substituídos por modelos simulados em tempo real. Inicialmente usado no teste de sistemas de travagem anti-bloqueio e sistemas de gestão e controlo de motores (ver exemplos em [Suh00], [Moskwa95] ou [Isermann99]), a HILS implantou-se na indústria automóvel à medida que o número de unidades de controlo electrónicas (ECUs) aumentou nos veículos. De facto, é hoje usada pelos principais construtores no teste e melhoramento de algoritmos de controlo para suspensões activas, sistemas de ABS, sistemas de transmissão automática ou para controlo de estabilidade dinâmica [Brennan 98]. Por exemplo, Otter em [Otter97] e [Otter00a], e Schlegel em [Schlegel02] apresentam modelos para sistemas de transmissão automáticos adequados à implementação de experiências HILS. Hwang, em [Hwang97], usa a HILS para o teste de algoritmos de controlo para suspensões activas.

Noutras áreas, a HILS foi usada com sucesso como uma ferramenta de desenvolvimento e avaliação de software/hardware de controlo numa barragem hidroeléctrica [Mansoor00]. Em [Ferreira99b] propõe-se uma metodologia para modelação de sistemas hidráulicos para HILS. Shetty, em [Shetty01], apresenta uma nova metodologia para o design de sistemas mecatrónicos baseada em HILS. Zlajpah, em [Zlajpah98], descreve uma biblioteca de modelos para teste de controladores de manipuladores planares usando HILS. Em [Engelbrecht99] é apresentada uma plataforma HILS para teste de algoritmos de controlo de tráfego automóvel.

### 6.3.4 Plataformas HILS implementadas

Neste trabalho, para simulação em tempo real e execução de experiências de HILS, recorreu-se a três plataformas diferentes: plataforma fornecida pela dSPACE, o xPC e o RTWT, fornecidos pela Mathworks. Duas dessas plataformas (dSPACE e RTWT) foram instaladas no mesmo computador, com o sistema operativo Windows NT 4.0 da Microsoft, Figura 6.32. A plataforma xPC foi instalada num outro computador pessoal com o sistema operativo Windows 2000. Em qualquer das três plataformas os modelos podem ser implementados na linguagem Modelica ou em Simulink. Os modelos desenvolvidos em linguagem Modelica são convertidos, através da interface Modelica-Simulink fornecida pela plataforma Dymola, num bloco, implementado por uma função em linguagem C (*s-function*), que pode ser usado num qualquer modelo Simulink.



## Plataforma dSPACE e RTWT

A Figura 6.32 mostra os blocos de hardware e software, bem como as suas interligações, para as plataformas dSPACE e RTWT. O Real-Time Workshop (RTW) gera automaticamente o código ANSI C, otimizado para execução em tempo real, a partir de modelos em Simulink. O código C é gerado por uma ferramenta chamada Target Language Compiler (TLC), que funciona como um processador de texto, a partir de uma forma intermédia do modelo em Simulink (.rtw) e dos ficheiros *target* (.tlc).

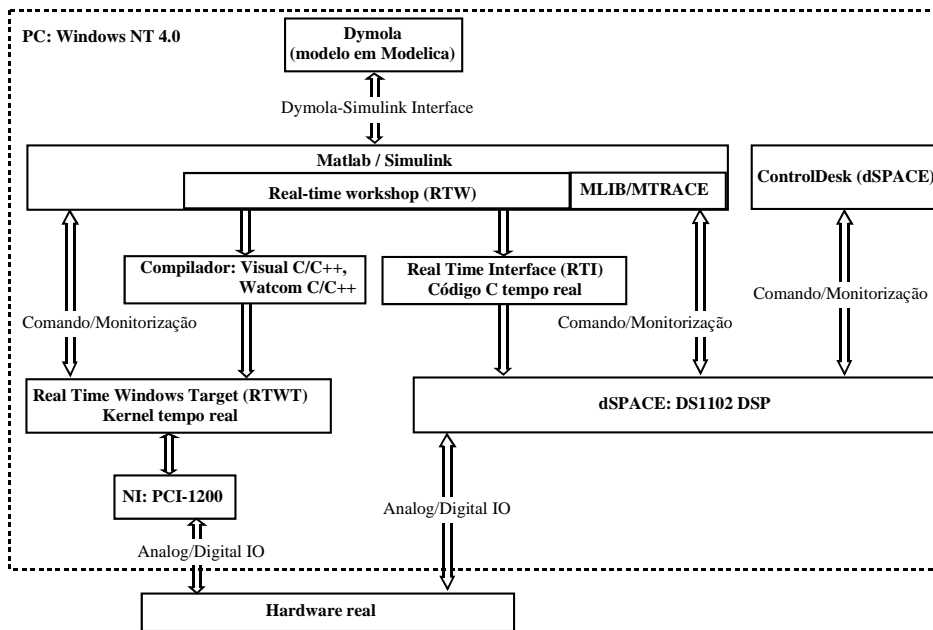


Figura 6.32 Plataforma implementada (dSPACE, RTWT) para a realização de experiências HILS

No caso da utilização da plataforma dSPACE, a Real Time Interface (RTI) expande o RTW com um conjunto de ferramentas que permitem compilar o código ANSI C gerado pelo RTW, incorporar as funções da dSPACE e carregar o programa executável no hardware de tempo real. A RTI gera também um ficheiro com as referências aos sinais e parâmetros definidos no modelo em Simulink, que permite o acesso à actualização de parâmetros e monitorização de variáveis através do Matlab (utilizando uma biblioteca de funções MLIB/MTRACE fornecida pela dSPACE). É também possível usar o ControlDesk para a interactividade com as aplicações tempo real, permitindo, além disso, a rápida construção de interfaces para experiências através de mecanismos *drag & drop*, já que disponibiliza um conjunto de instrumentos de visualização e de comando pré-definidos. O comando/monitorização das experiências ocorre sem interrupção da aplicação de tempo real. O uso das bibliotecas MLIB/MTRACE permite também realizar experiências de teste em sequência e, por exemplo, realizar tarefas de optimização de parâmetros de controladores através das capacidades de cálculo e de ferramentas fornecidas pelo Matlab.

A segunda opção consiste em utilizar um núcleo de tempo real, fornecido pelo *Real Time Windows Target* (RTWT), em conjunto com uma carta de aquisição de dados da National Instruments (PCI-1200). O código C gerado pelo RTW pode ser compilado pelo Visual C/C++ ou pelo Watcom C/C++ e carregado no núcleo de tempo real através de mecanismos fornecidos pelo RTW e Simulink. O comando e a monitorização obtêm-se através do Simulink, a funcionar em modo externo, ou pelo Matlab através de mecanismos de comunicação com o Simulink.

Em ambas as plataformas a comunicação com o hardware real é estabelecida através do sistema de entradas e saídas analógicas e digitais das respectivas cartas.

### A plataforma xPC

Os modelos são simulados em tempo real de forma determinística, num computador dedicado onde é instalado o núcleo de tempo real do xPC (Figura 6.33).

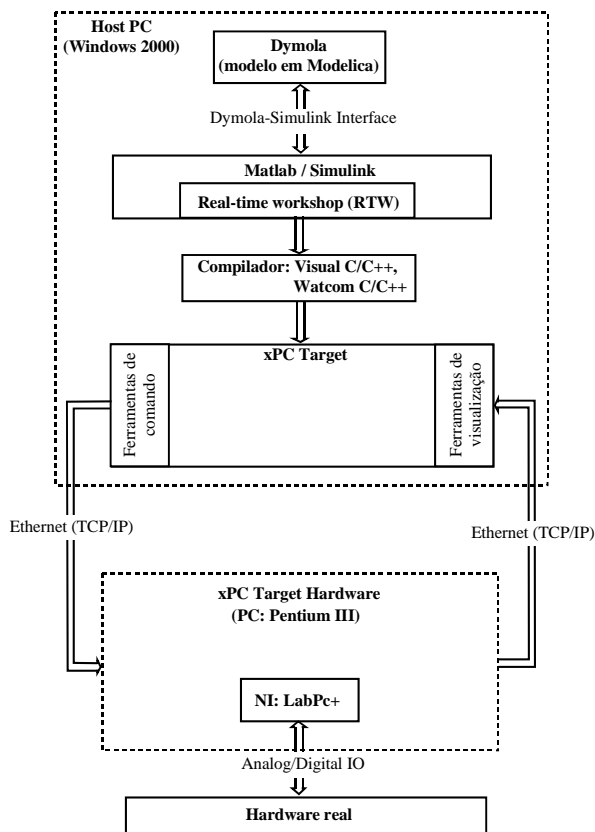


Figura 6.33 Plataforma para realização de HILS com o xPC

Neste computador, o xPC apenas usa a memória, não sendo necessário instalar qualquer software adicional. São fornecidas ferramentas, através de linhas de comando do Matlab (instalado no computador *host*), para controlar a execução tempo real dos modelos. As ferramentas de comando e de visualização permitem, por exemplo, alterar parâmetros do modelo ou seguir trajectórias de

variáveis em tempo real. A comunicação entre o Host PC e o Target PC é suportada pelo protocolo TCP/IP, permitindo rápidas taxas de transmissão de informação. No presente momento, a Mathworks não fornece *drivers* para as cartas disponíveis (LabPC+ e PCI 1200 da National Instruments), pelo que houve a necessidade de desenvolver *drivers* em Linguagem C (através de uma *s-function*) para poder usar a carta LabPC+. Todo o processo de compilação é idêntico ao usado na plataforma RTWT.

### 6.3.5 Experiências de HILS

Realizaram-se várias experiências, nas diferentes plataformas de HILS, com os modelos de componentes hidráulicos propostos no Capítulo 3.

#### 6.3.5.1 Experiências HILS na plataforma dSPACE

A Figura 6.34 ilustra uma das experiências realizadas na plataforma dSPACE. Neste caso, o exemplo é o controlo em malha fechada do eixo linear do SCARA (controlador proporcional com  $K_p = 50$ ), onde o sinal de referência (apresentado na Figura 6.38) é uma trajectória de posição para deslocamentos ponto a ponto. O cilindro é actuado na vertical com uma massa total de 80Kg. O sistema foi simulado em tempo real com um algoritmo de integração explícito de 3ª ordem, com um passo fixo e igual a  $0.5ms$ .

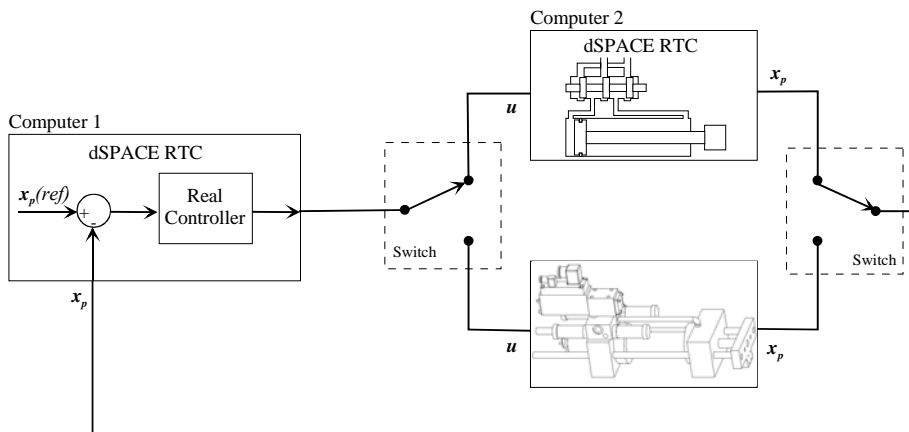


Figura 6.34 Experiência HILS através da plataforma dSPACE

O modelo em linguagem Modelica, apresentado a seguir, usa a biblioteca de modelos desenvolvida. O modelo é compilado e inserido numa *s-function* do Matlab podendo, assim, ser simulado em tempo real na carta DS1102. O bloco *DymolaBlock*, Figura 6.35, é o resultado da conversão de Modelica para Simulink do modelo *ServoSystemDSPACE*. A entrada para a válvula,  $u$ , foi

declarada *input* e as pressões nas câmaras do cilindro,  $P_1$  e  $P_2$ , a posição do êmbolo,  $x_p$ , e os estados do statechart, *Normal*, *StopLeft* e *StopRight*, são declarados *output*. Dessa forma é possível alterar a entrada para a válvula e monitorar as outras variáveis a partir do modelo Simulink. Note-se que os parâmetros do modelo (disponíveis no bloco *DymolaBlock*) também poderão ser alterados no arranque de cada simulação.

```

model ServoSystemDSPACE
  parameter Pressure pPump = 70e5;
  parameter Mass mCyl = 80;
  Hydraulic.HydPump.IdealPumps.CtePressurePump P(ctePressure=pPump);
  Hydraulic.HydBasic.HydElements.HydTank T;
  Hydraulic.HydValve.ServoSolenoidValve.ValveModeloEstatico11 pValve;
  Hydraulic.HydCyl.SingleRodCyl.HydCylinderHardStop hAct(M = mCyl);
  input Real u;
  output Pressure P1,P2;
  output Position xp;
  output Boolean Normal, StopLeft, StopRight;
equation
  connect(pValve.HydS, P.HydA);
  connect(pValve.HydT, T.HydA);
  connect(pValve.HydA, hAct.HydA);
  connect(pValve.HydB, hAct.HydB);
  pValve.xs = u;
  P1 = hAct.P1;
  P2 = hAct.P2;
  xp = hAct.xp;
  Normal = hAct.Normal;
  StopLeft = hAct.StopLeft;
  StopRight = hAct.StopRight;
end ServoSystemDSPACE;

```

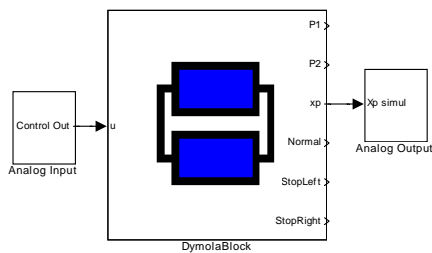


Figura 6.35 Modelo Simulink usado na experiência de HILS

O valor de referência de posição ultrapassa, propositadamente, os limites físicos para o curso do êmbolo do cilindro. Na Figura 6.36 verifica-se, através da actividade dos estados *StopLeft* e *StopRight*, que a posição do êmbolo não acompanha o sinal de referência.

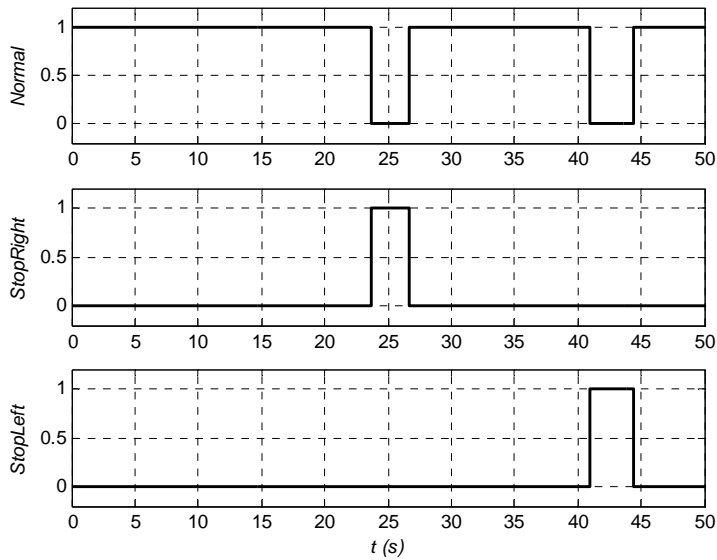


Figura 6.36 Diagrama temporal representando o estado do êmbolo do cilindro. Ver statechart na secção 4.2

Os sinais de pressão nas câmaras do cilindro são apresentados na Figura 6.37. Verifica-se que os valores das pressões nas câmaras tendem para as pressões da fonte e do tanque, na altura do embate do êmbolo nos limites de curso.

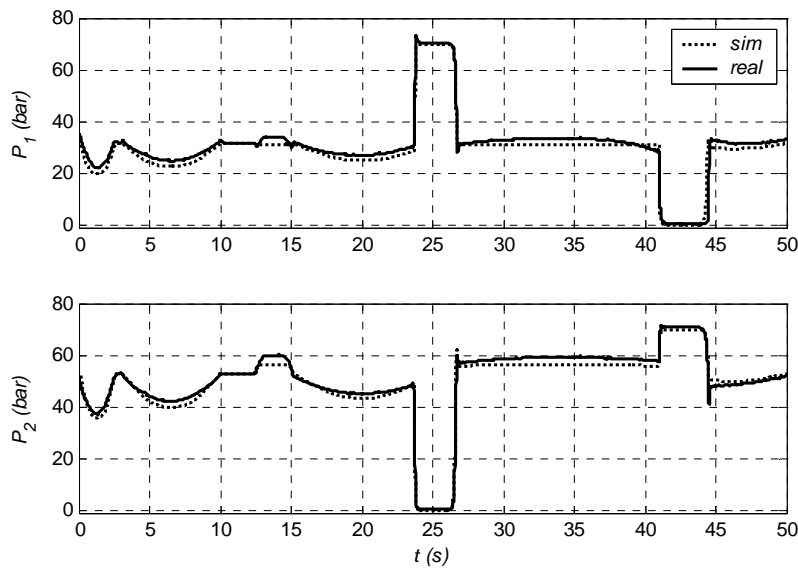


Figura 6.37 Pressões nas duas câmaras do cilindro.

Os sinais de referência de posição, das posições real e simulada do êmbolo, assim como dos erros de posição simulados e reais, são apresentados nas figuras 6.38, 6.39 e 6.40. De salientar o pormenor (e a ampliação) dos sinais quando o êmbolo atinge o limite de curso.

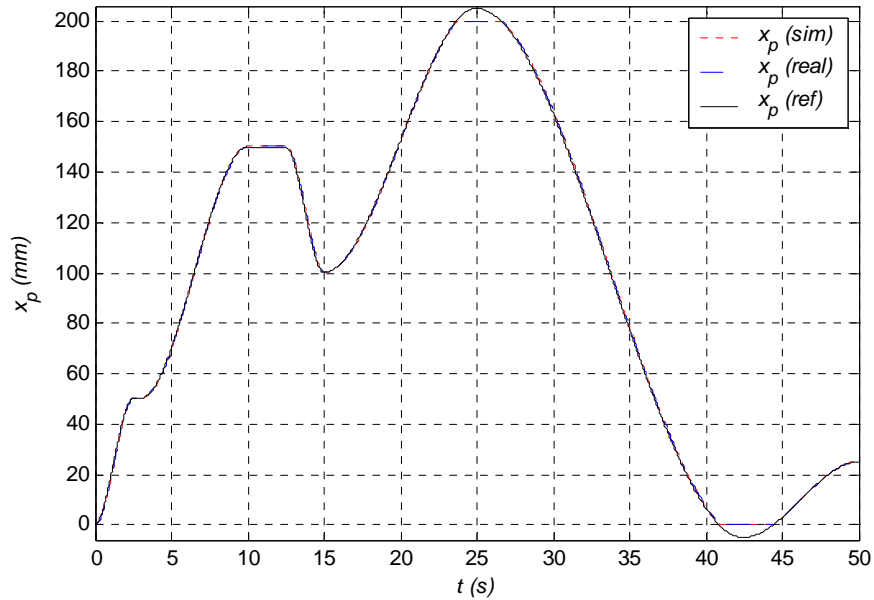


Figura 6.38 Sinal de posição de referência, da posição real e simulado do êmbolo do cilindro

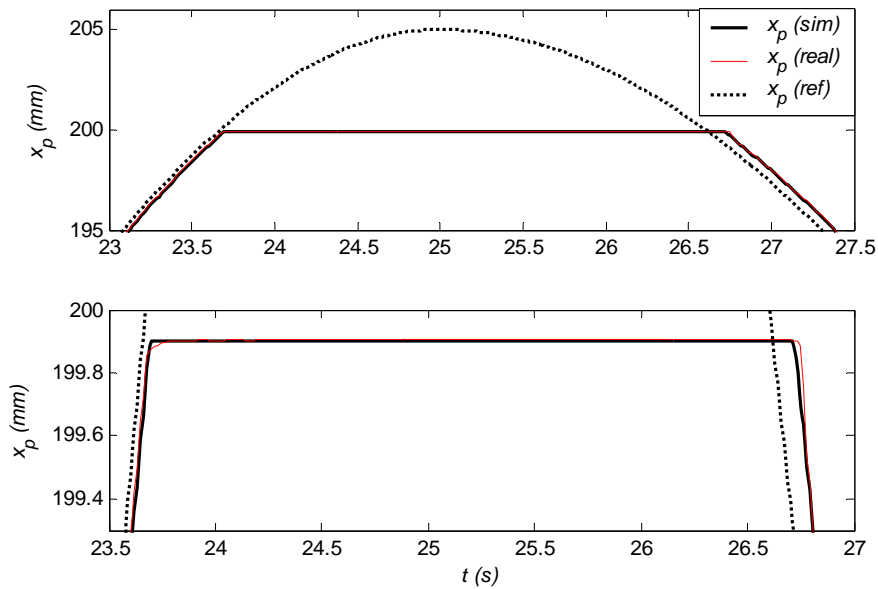


Figura 6.39 Pormenor (e ampliação) dos sinais de posição (referência, real e simulada) na zona de limite de curso

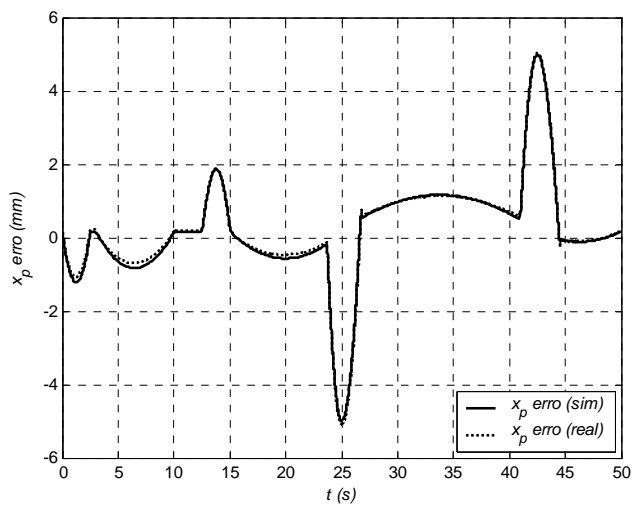


Figura 6.40 Erro de posição real e simulado

Nas mesmas condições da experiência anterior realizaram-se outras em que se procurou evidenciar o desempenho do modelo, representado na Figura 6.35, em condições dinâmicas mais exigentes. Nestas experiências, as linhas de transmissão da fonte para a válvula e da válvula para o tanque, foram modeladas através de uma condutância igual a  $5.7 \cdot 10^{-10} m^3 s^{-1} Pa^{-1}$ , tendo sido obtidos resultados satisfatórios na comparação das pressões na entrada e saída da válvula, Figura 6.42.

A experiência, apresentada nas figuras 6.41, 6.42 e 6.43, procurou avaliar o desempenho dos sistemas, real e simulado, a entradas em rampa e em degrau. Esta foi realizada para uma pressão na fonte  $P_s = 120bar$ , e um ganho proporcional para o controlador de  $K_p = 100$ .

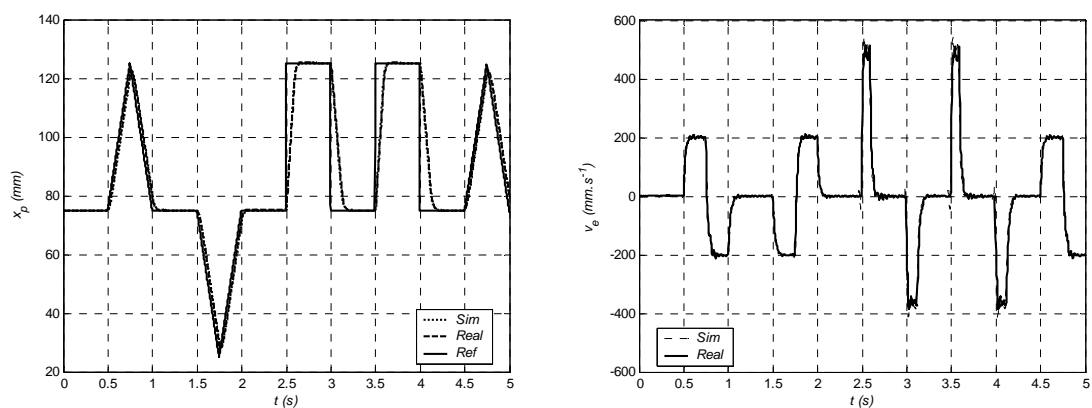


Figura 6.41 Sinais de posição (referência, real e simulada) e velocidade (real e simulada) do êmbolo do cilindro

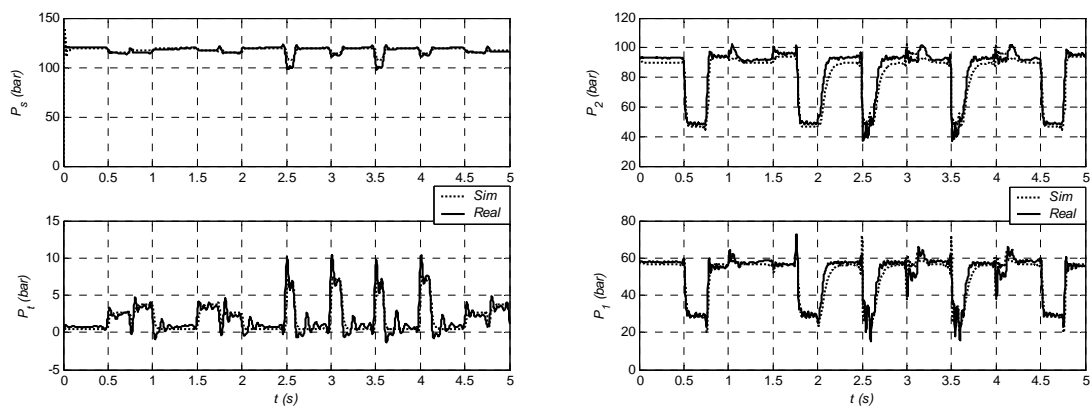


Figura 6.42 Pressões nos terminais da válvula e nas câmaras do cilindro (reais e simuladas)

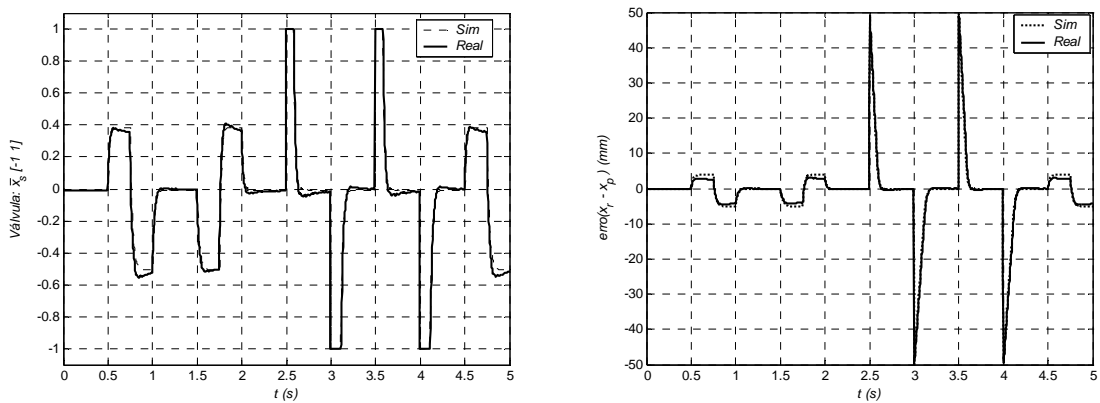


Figura 6.43 Sinais enviados para a válvula e erros de posição (reais e simulados)

Outra experiência, cujos resultados são ilustrados nas figuras 6.44, 6.45 e 6.46, explorou a resposta do sistema a uma referência sinusoidal ( $P_s = 120\text{bar}$ ,  $K_p = 50$ ).

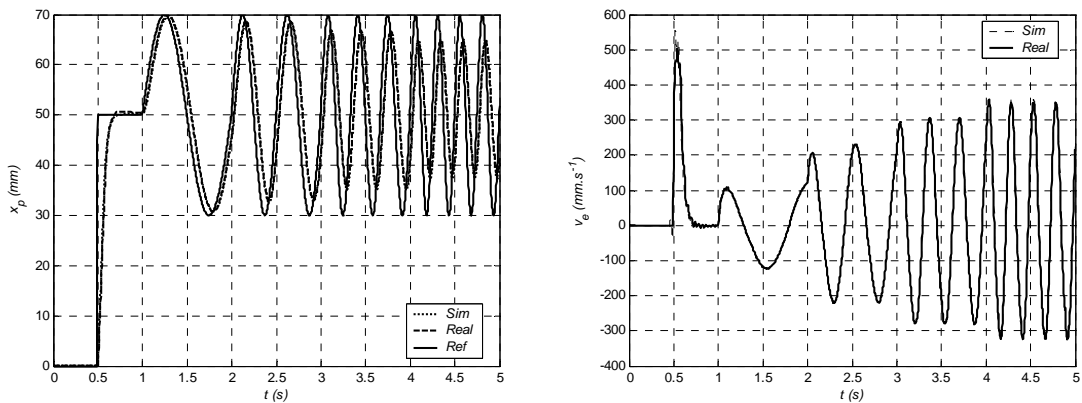


Figura 6.44 Sinais de posição (referência, real e simulada) e velocidade (real e simulada) do êmbolo do cilindro



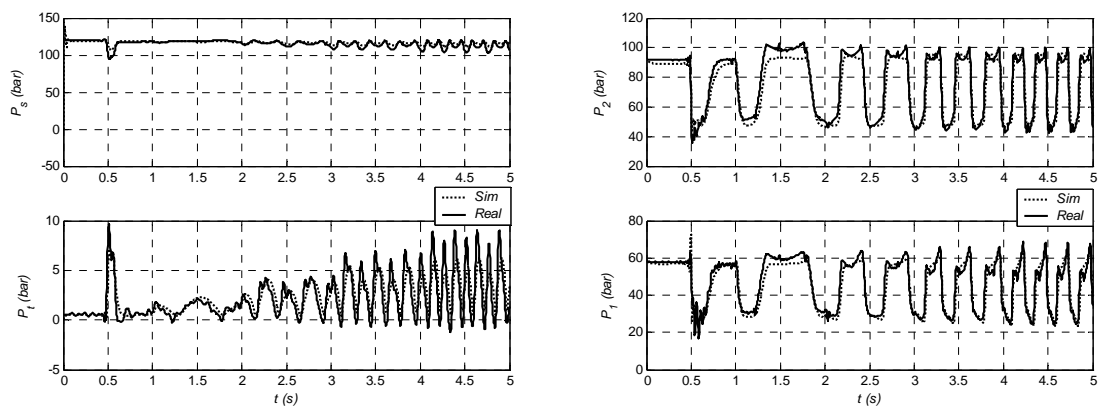


Figura 6.45 Pressões nos terminais da válvula e nas câmaras do cilindro (reais e simuladas)

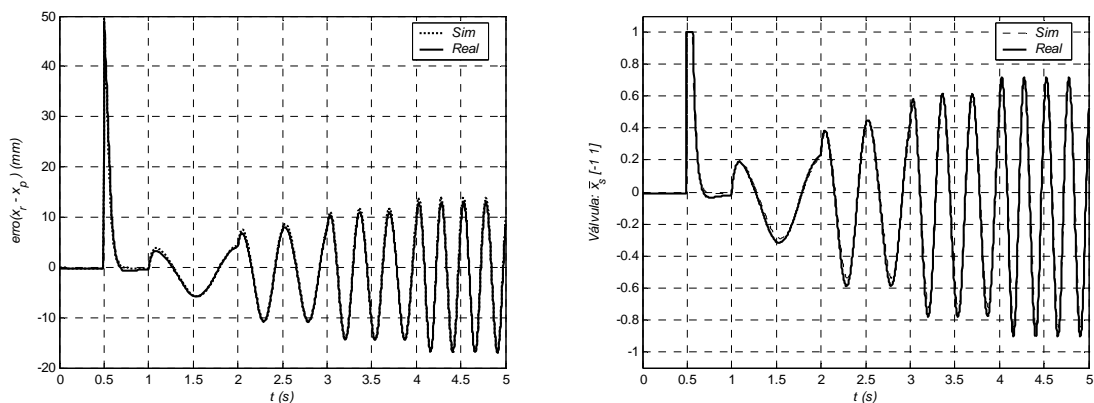


Figura 6.46 Sinais enviados para a válvula e erros de posição (reais e simulados)

Pela análise dos resultados, ilustrados nas três experiências anteriores, pode-se concluir que o modelo do conjunto, válvula + cilindro, apresenta um bom desempenho, quer em regime de baixas velocidades, quer para trajectórias de elevada dinâmica.

### 6.3.5.2 Experiências HILS na plataforma RTWT

Nesta secção apresenta-se um exemplo de utilização dos modelos desenvolvidos, na modelação de uma suspensão activa, para uso em experiências de HILS na plataforma RTWT (Figura 6.32).

Modelos de um “quarto de veículo” são muito comuns na análise de suspensões, pois são simples e capturam muitas das características importantes do modelo completo [Lin97]. A Figura 6.47 mostra um modelo de uma suspensão activa (1/4 de carro) com um actuador hidráulico ligado em paralelo com um conjunto passivo mola/amortecedor. O modelo simplificado, muitas vezes usado para modelação, pode ver-se na Figura 6.48.

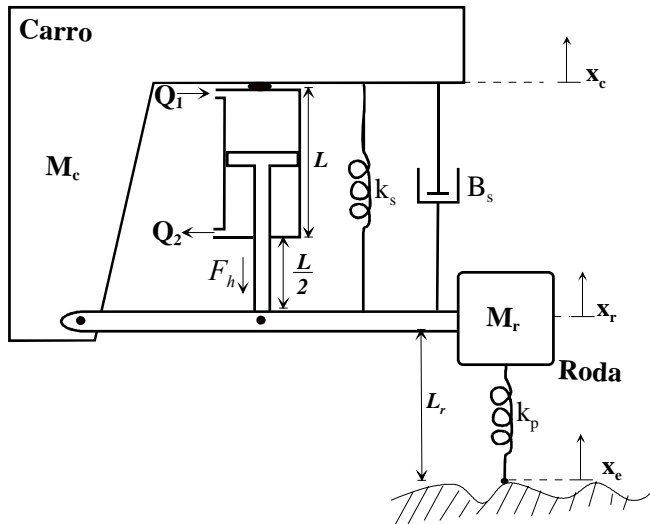


Figura 6.47 Modelo de uma suspensão activa (¼ de carro) com actuador hidráulico ligado em paralelo com um conjunto passivo mola/amortecedor

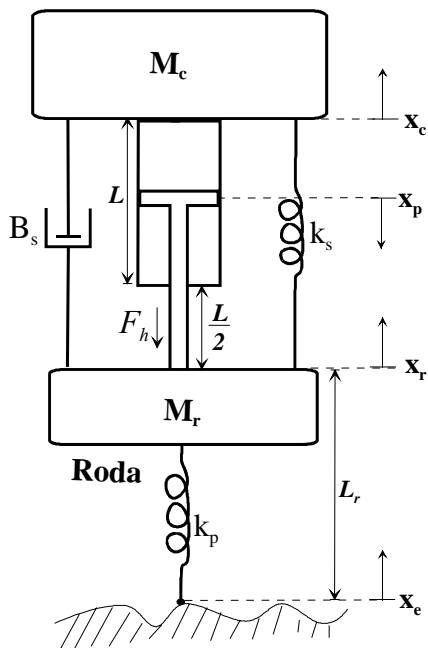


Figura 6.48 Modelo simplificado da suspensão activa

As equações de movimento para o modelo da Figura 6.48 são:

$$M_c \ddot{x}_c + B_s \dot{x}_c + K_s x_p - F_h = 0 \quad (6.26)$$

$$M_r (\ddot{x}_c - \ddot{x}_p) - B_s \dot{x}_p - K_s x_p + K_s \left( x_c - x_p - x_e - 3\frac{L}{2} - L_r \right) + F_h = 0 \quad (6.27)$$

$$x_r = x_c - x_p - 3\frac{L}{2} \quad (6.28)$$

em que  $M_c = 290Kg$  representa a massa de ¼ de carro mais a massa do corpo do cilindro,  $M_r = 59Kg$  representa a massa do conjunto roda, pneu, êmbolo e haste do cilindro,  $K_s = 16812N / m$  é a rigidez da suspensão,  $B_s = 1000Ns / m$  é o coeficiente de amortecimento da suspensão e  $K_p = 190000N / m$  representa a rigidez do pneu. A variável  $x_e$  é o distúrbio da suspensão activa e representa as possíveis oscilações na estrada. A posição intermédia do cilindro corresponde a  $x_p = 0$  e a posição da roda  $x_r$  pode ser avaliada através de (6.28). A posição do carro,  $x_c$ , deve ser inicializada a  $3\frac{L}{2} + L_r$ .

Nesta experiência usou-se, para a válvula de controlo, o modelo estático 5 (estrutura fixa) e o modelo dinâmico 11, e um modelo para o cilindro com a posição inicial na zona intermédia. A simulação com passo fixo, devido à rigidez do sistema de equações, exigia passos de integração muito pequenos (da ordem dos  $100ms$ ) para algoritmos de 3ª ordem, o que não permitia a simulação em tempo real. Para a simulação em tempo real usaram-se então os modos de integração, relatados na secção 6.3.1, denominados *inline integration* e *mixed mode integration*, disponíveis na versão mais recente da plataforma Dymola (versão 4.2a). A utilização destes modos implica a utilização do método de Euler de 1ª ordem e a definição das variáveis de estado que contribuem com as menores constantes de tempo. No presente caso, consideraram-se as pressões nas câmaras do cilindro, *Cil.P1* e *Cil.P2*, como sendo os estados que, em termos numéricos, variam mais rapidamente. Para usar estes métodos é necessário definir um outro modelo, *SuspensionRTWT*, que expande o modelo *Suspension* com a *String fastStates*, que indica ao compilador quais os estados “rápidos”.

```

model SuspensionRTWT
  extends Suspension;
  String[:] fastStates={"Cil.P1","Cil.P2"};
end SuspensionRTWT;

```

A Figura 6.49 mostra o modelo em Simulink para a realização da experiência HILS, de controlo da suspensão activa na plataforma RTWT. O sinal de distúrbio da estrada,  $x_e$ , foi introduzido através de uma LUT (*Look Up Table*). O módulo *Adaptação do sinal* converte os níveis de tensão adquiridos na carta LabPC+ em valores relativos compatíveis com os sinais de entrada do modelo da válvula.

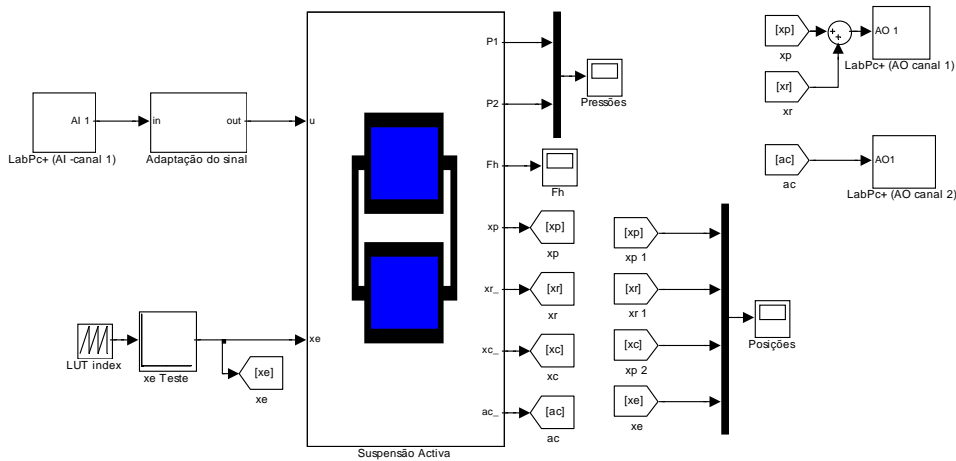


Figura 6.49 Modelo Simulink para realização da experiência HILS de controlo da suspensão na plataforma RTWT

Para o teste do modelo *SuspensionRTWT*, em experiências de HILS, implementou-se um controlador “real”, em hardware, através de três amplificadores operacionais (Figura 6.50). Pretende-se obter a compensação dos deslocamentos da roda,  $x_r$ , através do deslocamento do êmbolo do cilindro recorrendo a um controlador PD (proporcional + derivativo), e de uma atenuação da aceleração do carro,  $a_c$ , com um controlador proporcional. O uso de controladores mais sofisticados, de que são exemplos os trabalhos de [Istepanian96], [Lin97], [Peng97] ou [Ikenaga99], está fora do âmbito deste trabalho.

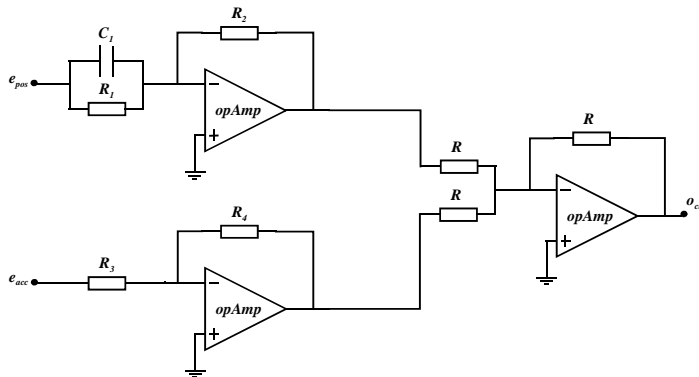


Figura 6.50 Controlador de teste da suspensão activa implementado com amplificadores operacionais

Considerando todos os componentes ideais, a saída do controlador da Figura 6.50 é dada por:

$$O_{ctr} = \frac{R_2}{R_1} e_{pos} + R_1 C \frac{de_{pos}}{dt} + \frac{R_4}{R_3} e_{acc} \quad (6.29)$$

onde  $e_{pos} = - (x_p + x_r)$  e  $e_{acc} = - a_c$  são, respectivamente, o erro de posição e o erro de aceleração.

Os parâmetros do controlador são então:

$$k_p = \frac{R_2}{R_1}, k_d = R_1 C \text{ e } k_{pac} = \frac{R_4}{R_3} \quad (6.30)$$

O modelo foi simulado, com um passo de integração fixo de  $1ms$ , num PC com processador PentiumIII a  $1GHz$ . A percentagem de uso do processador pelo núcleo do RTWT, na simulação do modelo, foi de 19% da capacidade de processamento. Os resultados da experiência HILS para  $k_p = 20$ ,  $k_d = 10$ ,  $k_{pac} = 2$  e pressões na fonte de  $P_s = 140bar$  e  $P_s = 70bar$  são apresentados nas figuras 6.51 a 6.54. As posições são apresentadas simultaneamente para uma melhor visibilidade do desempenho da suspensão activa.

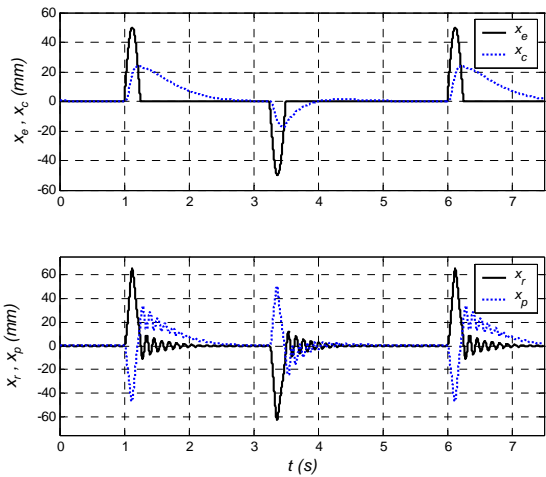


Figura 6.51 Posições obtidas para  $k_p = 20$ ,  $k_d = 10$  e  $k_{pac} = 2$  e a pressão da fonte  $P_s = 140bar$

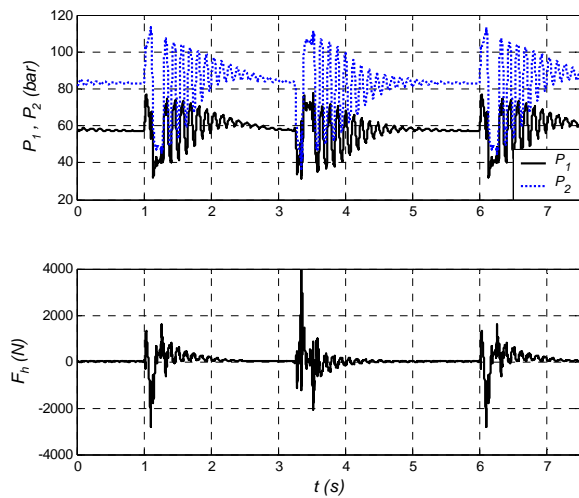


Figura 6.52 Pressões nas câmaras e força exercida pelo cilindro hidráulico ( $P_s = 140bar$ )

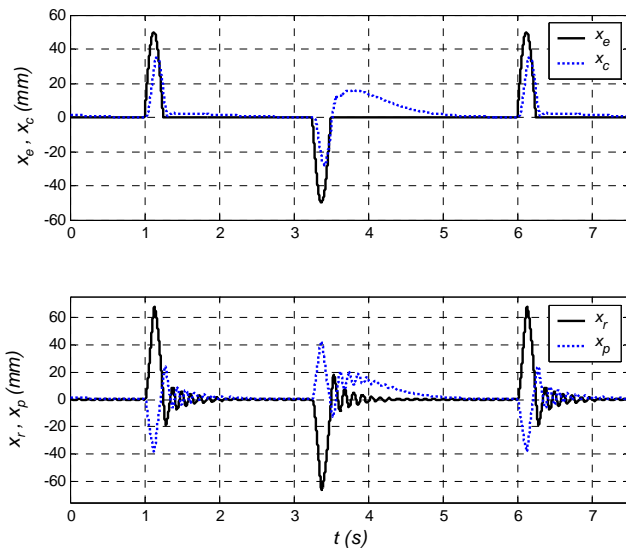


Figura 6.53 Posições obtidas na experiência HILS da Figura 6.49 para  $k_p = 20$ ,  $k_d = 10$  e  $k_{pac} = 2$  e a pressão da fonte  $P_s = 70bar$

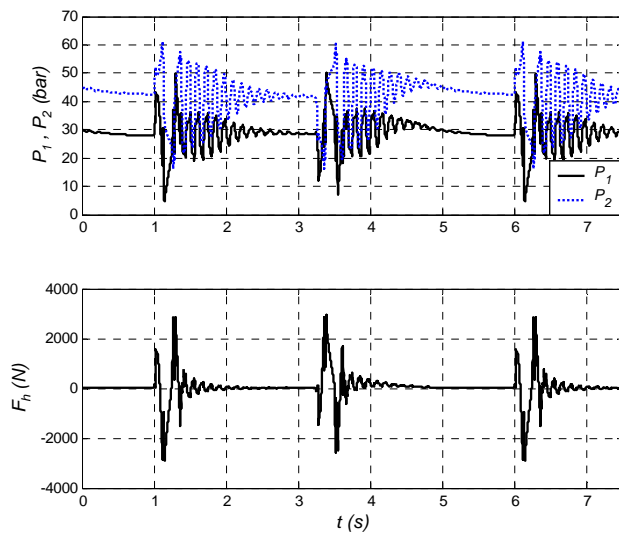


Figura 6.54 Pressões nas câmaras e força exercida pelo cilindro hidráulico ( $P_s = 70bar$ )

### 6.3.5.3 Experiências HILS na plataforma xPC

Nesta secção apresenta-se um exemplo de utilização dos modelos desenvolvidos na modelação do manipulador hidráulico (vide secção 6.1), para uso em experiências de HILS na plataforma xPC (Figura 6.33).

## Modelo dinâmico do manipulador

No caso do presente manipulador, a dinâmica do eixo prismático pode ser considerada desacoplada da dinâmica dos dois braços com movimento angular. Somente a sua distribuição de massa (actuador + carga) influencia a matriz de inércia, através da variação de massa e do centro de massa do braço 2. Sendo assim, a sua dinâmica pode ser descrita separadamente. Tem-se, então, o caso de um manipulador com dois graus de liberdade (Figura 6.55) com juntas rotativas.

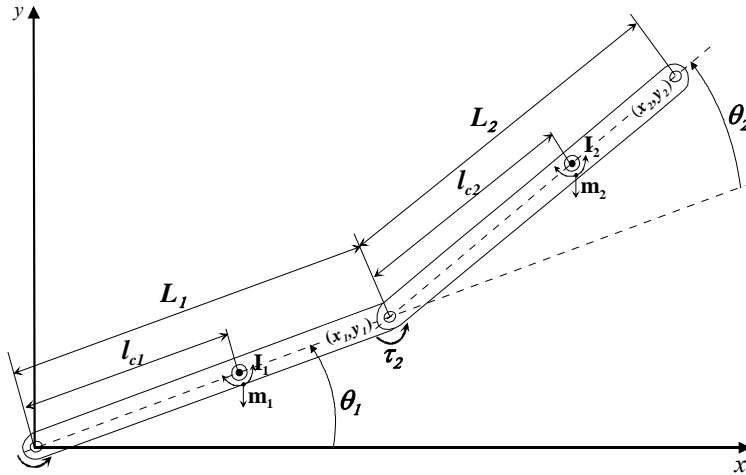


Figura 6.55 Manipulador com dois graus de liberdade

Na Figura 6.55 considera-se:

$l_{c1}, l_{c2}$  – distâncias do centro de massa às juntas respectivas

$L_1, L_2$  – comprimento dos braços

$m_1, m_2$  – massa total dos braços ( $m_2$  engloba toda a massa do braço 2, do eixo prismático e da carga)

$I_1, I_2$  – momentos de inércia dos braços referidos aos seus centros de massa

De notar que a variação da carga implica a variação de  $m_2, I_2$  e  $l_{c2}$ . Devido à complexidade geométrica dos braços do actuador, os momentos de inércia e os centros de massa foram extraídos do modelo 3D (Figura 6.2) através de funcionalidades oferecidas pelo software de CAD utilizado. Para o cilindro sem carga os valores extraídos foram:

$$m_1 = 27.0Kg, I_1 = 0.675Kgm^2, l_{c1} = 0.31m, L_1 = 0.500m$$

$$m_2 = 33.2Kg, I_2 = 1.027Kgm^2, l_{c2} = 0.32m, L_2 = 0.417m$$

Através da aplicação do método de Lagrange, e considerando a energia potencial nula, os binários motores aplicados nas juntas podem ser expressos por [Almeida93]:

$$t = J(q)\ddot{q} + c(q, \dot{q}) \quad (6.31)$$

onde  $\theta$  é o vector de posições das juntas,  $q = \begin{bmatrix} q_1 \\ q_2 \end{bmatrix}$ ,  $J$  é a matriz de inércia do manipulador,  $c$  é o vector dos binários introduzidos pelas forças de Coriolis e centrífugas, e  $\tau$  é o vector dos binários aplicados às juntas,  $t = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix}$ .

Os componentes de  $\tau$  são expressos por:

$$t_1 = J_{11}\ddot{q}_1 + J_{12}\ddot{q}_2 + h_1 \quad (6.32)$$

$$t_2 = J_{21}\ddot{q}_1 + J_{22}\ddot{q}_2 + h_2 \quad (6.33)$$

onde

$$J_{11} = m_1 l_{c1}^2 + I_1 + m_2 (L_1^2 + l_{c2}^2 + 2L_1 l_{c2} \cos(q_2)) + I_2 \quad (6.34)$$

$$J_{12} = J_{21} = m_2 L_1 l_{c1} \cos(q_2) + m_2 l_{c2}^2 + I_2 \quad (6.35)$$

$$J_{22} = m_2 l_{c2}^2 + I_2 \quad (6.36)$$

$$h_1 = -2\dot{q}_1 \dot{q}_2 m_2 L_1 l_{c2} \sin(q_2) - \dot{q}_2^2 m_2 L_1 l_{c2} \sin(q_2) \quad (6.37)$$

$$h_2 = m_2 L_1 l_{c2} \sin(q_2) \dot{q}_1^2 \quad (6.38)$$

As acelerações angulares das juntas são dadas por:

$$\ddot{q}_1 = \frac{J_{22}(t_1 - h_1) - J_{12}(t_2 - h_2)}{J_{11}J_{22} - J_{12}J_{21}} \quad (6.39)$$

$$\ddot{q}_2 = \frac{-J_{21}(t_1 - h_1) + J_{11}(t_2 - h_2)}{J_{11}J_{22} - J_{12}J_{21}} \quad (6.40)$$

para  $J_{11}J_{22} - J_{12}J_{21} \neq 0$

O modelo expresso por (6.32) a (6.38) não incorpora a dinâmica dos actuadores (incluindo o atrito) e é válido para qualquer tipo de actuação. No presente caso (actuação hidráulica), os binários motores  $\tau_1$  e  $\tau_2$  são calculados pelos modelos dos actuadores hidráulicos.

## Modelo cinemático

As cinemáticas directa e inversa para este tipo de manipuladores são bastante simples:

Recorrendo à Figura 6.55, a cinemática directa é expressa por:

$$x_1 = L_1 \cos(q_1) \quad (6.41)$$



$$y_1 = L_1 \sin(q_1) \quad (6.42)$$

$$x_2 = L_1 \cos(q_1) + L_2 \cos(q_1 + q_2) \quad (6.43)$$

$$y_2 = L_1 \sin(q_1) + L_2 \sin(q_1 + q_2) \quad (6.44)$$

onde  $(x_1, y_1)$  e  $(x_2, y_2)$  representam a posição das extremidades do braço 1 e braço 2 relativamente à origem.

A cinemática inversa torna-se necessária para a definição de trajectórias do manipulador. Neste caso existe uma redundância, já que é possível atingir um determinado ponto no espaço de trabalho através de duas configurações possíveis.

$$\begin{cases} \dot{q}_1 = \text{atan2}(k_1, k_2) - \text{atan2}(\sqrt{k_1^2 + k_2^2 - k_3^2}, k_3) \\ \dot{q}_2 = \text{atan2}(\sqrt{k_2^2 - k_3^2}, k_3) \end{cases} \quad (6.45)$$

ou

$$\begin{cases} \dot{q}_1 = \text{atan2}(k_1, k_2) + \text{atan2}(\sqrt{k_1^2 + k_2^2 - k_3^2}, k_3) \\ \dot{q}_2 = - \text{atan2}(\sqrt{k_2^2 - k_3^2}, k_3) \end{cases} \quad (6.46)$$

em que:

$$\begin{cases} \dot{k}_1 = 2 \times L_1 \times y_2 \\ \dot{k}_2 = 2 \times L_1 \times x_2 \\ \dot{k}_3 = x_2^2 + y_2^2 + L_1^2 - L_2^2 \\ \dot{k}_{22} = 2 \times L_1 \times L_2 \\ \dot{k}_{33} = x_2^2 + y_2^2 - L_1^2 - L_2^2 \end{cases} \quad (6.47)$$

### Modelo do SCARA com actuação hidráulica

O circuito hidráulico modelado corresponde ao apresentado na Figura 6.3, onde se considerou uma pressão de trabalho constante,  $P_s = 140\text{bar}$ . As válvulas foram modeladas com o modelo estático 5 (estrutura fixa) e o modelo dinâmico 11. As válvulas  $V_1$  (Bosch de 24 l/min de caudal nominal),  $V_2$  (Bosch de 12l/min de caudal nominal) e  $V_3$  (Vickers de 24l/min de caudal nominal) foram parametrizadas através de valores obtidos experimentalmente, e de acordo com o especificado para o modelo estático e dinâmico. Os modelos para os actuadores angulares incorporam limites rígidos, especificados de forma semelhante ao statechart usado para os limites do cilindro (ver secção 3.3.3), e usam o modelo de atrito de LuGre simétrico. O modelo simulado da Figura 6.56 foi, então, usado para realização de experiências HILS na plataforma xPC. As trajectórias de referência para

os ângulos  $q_1$  e  $q_2$ , e para a posição do eixo linear  $x_p$ , foram geradas através de polinómios de 3º grau, típicos na especificação de trajectórias para movimentos ponto a ponto, e inseridas no modelo através de *Look Up Tables*. Os erros de trajectória são então enviados ao controlador real através de duas cartas de aquisição *LabPc+* (em virtude das cartas só possuírem duas saídas analógicas). Os controladores foram implementados em hardware de forma idêntica à implementação dos controladores descritos na secção 6.3.5.2. Neste caso, foram somente usados controladores proporcionais, com ganho 50 para a posição linear e ganho unitário para as posições angulares.

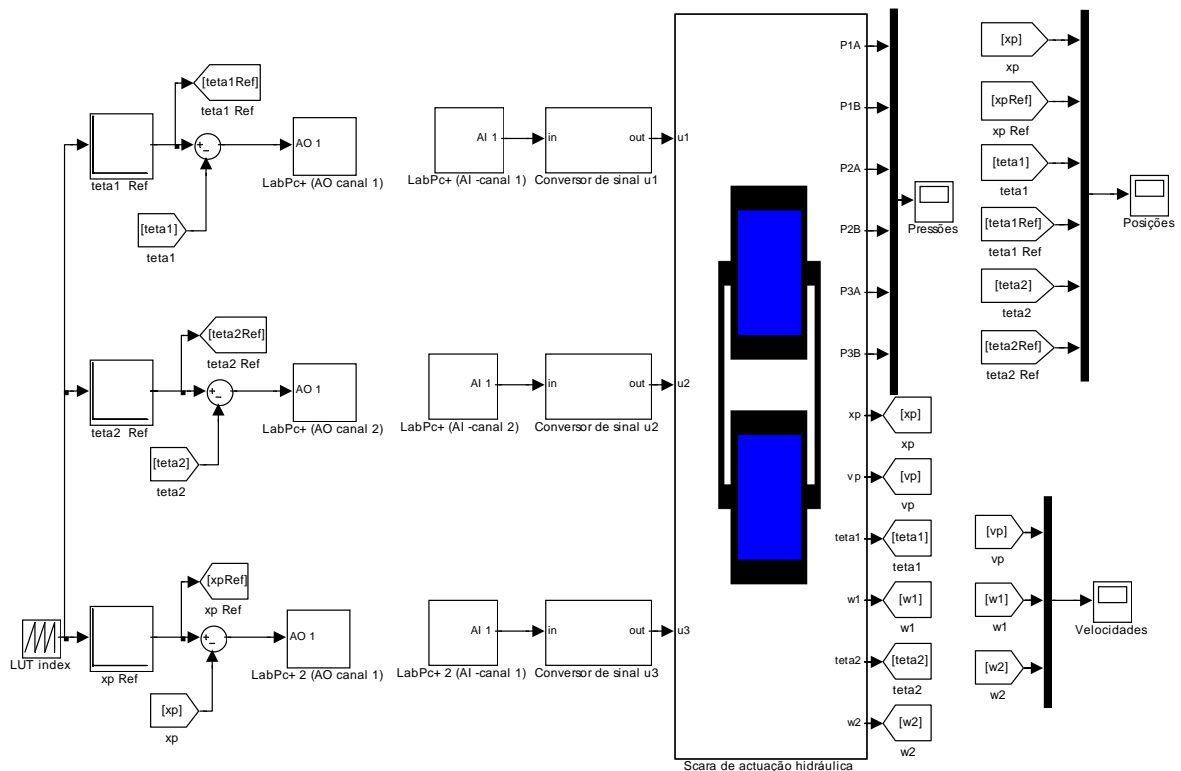


Figura 6.56 Modelo Simulink do SCARA hidráulico para realização de experiências HILS na plataforma xPC

O modelo foi simulado com o algoritmo de Euler de 1ª ordem e com um passo de integração de 1ms. Neste caso, também se recorreu a técnicas de *inline* e *mixed mode integration*, tendo sido todas as pressões declaradas como estados rápidos (*fastStates*). Nas figuras 6.57, 6.58 e 6.59 ilustram-se os resultados de uma das experiências, a qual consistiu numa sequência de movimentos durante um tempo de 50 segundos e com as trajectórias desejadas a exceder propositadamente os limites dos actuadores:  $[-140^\circ, 140^\circ]$  para os actuadores angulares dos braços 1 e 2, e  $[0, 200\text{mm}]$  para o eixo linear. Note-se que  $u_1$ ,  $u_2$  e  $u_3$  são os sinais de entrada das válvulas e  $w_1$  e  $w_2$  são as velocidades angulares das juntas 1 e 2, respectivamente.

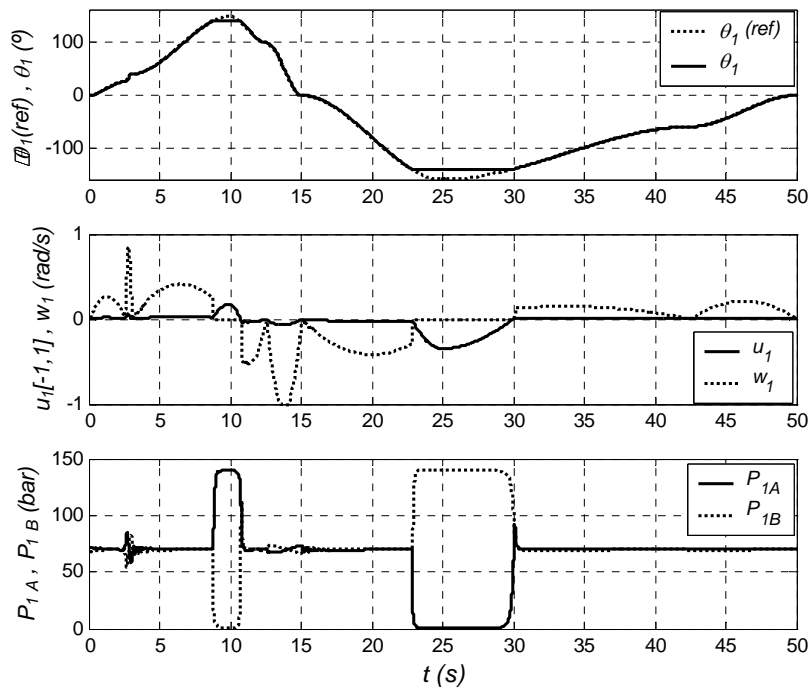


Figura 6.57 Variáveis correspondentes ao braço 1 do SCARA

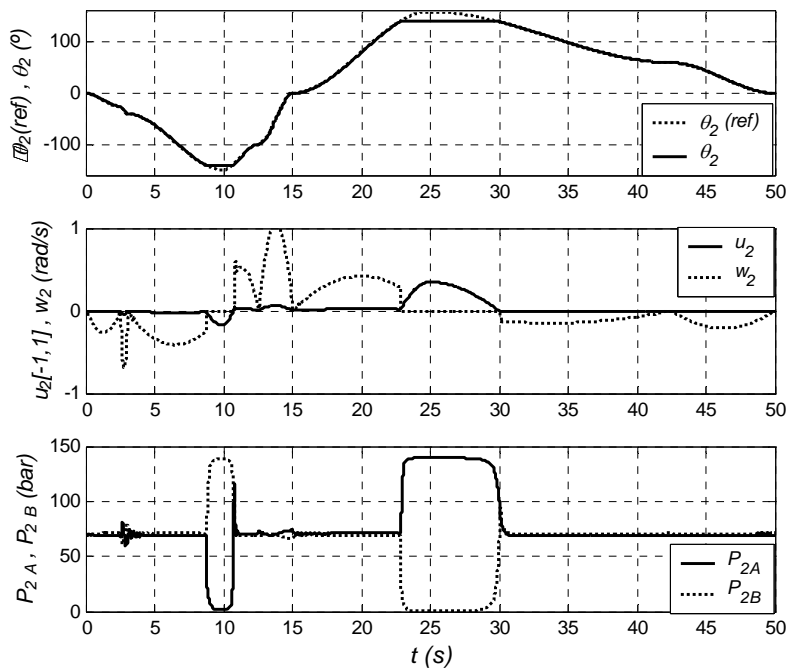


Figura 6.58 Variáveis correspondentes ao braço 2 do SCARA

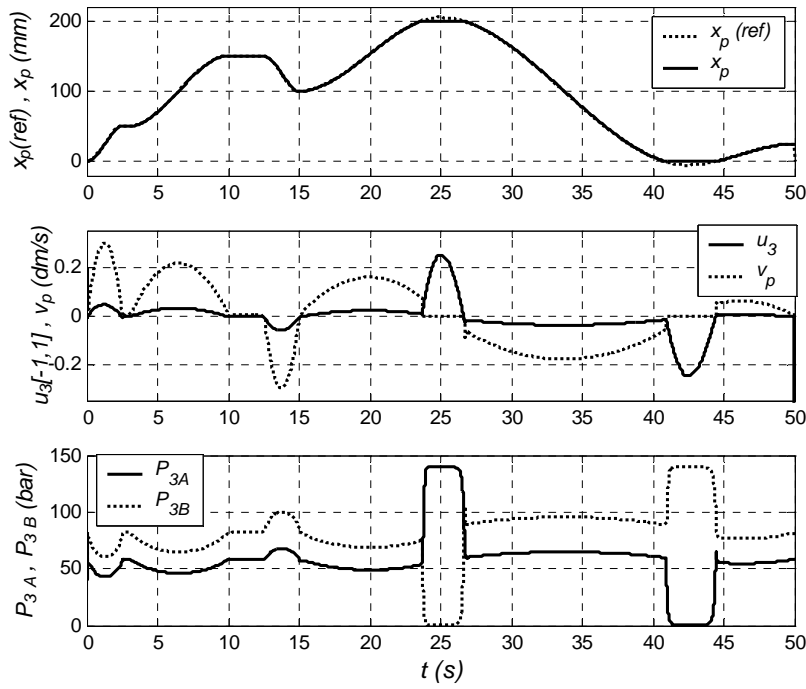


Figura 6.59 Variáveis correspondentes ao eixo linear do SCARA

Nas figuras 6.60 e 6.61 mostra-se uma ampliação dos sinais referentes aos dois braços do SCARA.

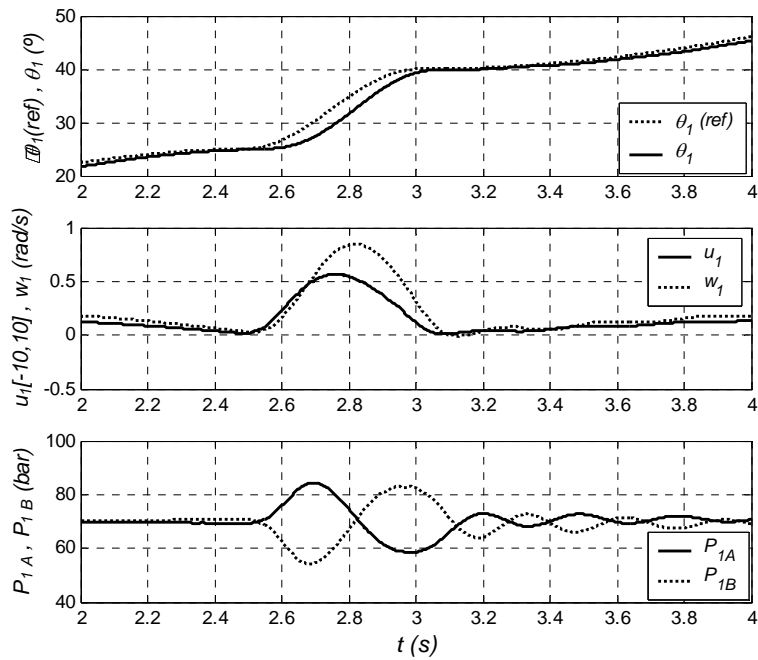


Figura 6.60 Variáveis correspondentes ao braço 1 do SCARA (ampliação)

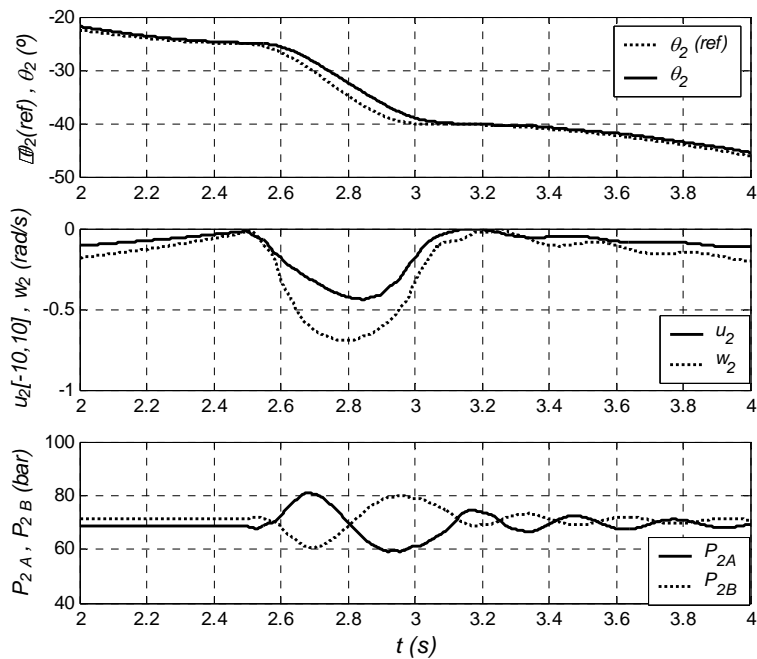


Figura 6.61 Variáveis correspondentes ao braço 2 do SCARA (ampliação)

Nesta secção relataram-se algumas experiências de HILS realizadas em plataformas de baixo custo. Alguns modelos apresentados nos Capítulos 4 e 5 foram simulados em tempo real, recorrendo a métodos de integração explícitos e mistos (implícitos e explícitos). Em todas as experiências realizadas com componentes hidráulicos foram usadas as pressões nas câmaras dos actuadores como as variáveis a integrar, recorrendo ao método implícito de Euler (declaradas como *fastStates* nos modelos em Modelica). A utilização dos métodos mistos permitiu passos de integração maiores e maior robustez nas experiências HILS. De referir também a utilização de controladores implementados em hardware, bastantes simples (proporcionais ou proporcionais + derivativos). Contudo, as plataformas desenvolvidas permitem o teste e parametrização de controladores mais sofisticados mas que, como já foi referido, estão fora do âmbito deste trabalho.

## 7 CONCLUSÕES FINAIS

O trabalho apresentado nesta dissertação teve como objectivo o desenvolvimento de modelos de diferentes complexidades, para componentes de sistemas hidráulicos, bem como a proposta de uma metodologia de modelação e organização de bibliotecas de modelos, com a finalidade de facilitar a gestão destes na realização de experiências de simulação com *hardware-in-the-loop*.

O conceito subjacente à metodologia de modelação proposta considera que um modelo de um componente é composto por duas perspectivas complementares que são a sua estrutura e o seu comportamento. Para tal, a metodologia toma por base a associação de uma linguagem de modelação orientada por objectos, chamada Modelica, para a modelação da estrutura, com o formalismo dos Statecharts, para a descrição do comportamento dinâmico do modelo. A metodologia de organização de bibliotecas de modelos permite a existência de modelos com diferentes complexidades para um mesmo componente. Deste modo, a realização de diferentes experiências de simulação (simulação em tempo não real ou em tempo real) é facilitada através da selecção da complexidade dos modelos mais adequada à simulação em causa.

O uso do formalismo dos Statecharts para a descrição de comportamentos complexos ganhou importância nos últimos anos e, com a introdução de características híbridas, descrevendo actividades contínuas dentro de estados, abre novas perspectivas na descrição de comportamentos de sistemas híbridos complexos. Este trabalho contribui para o alargamento destas perspectivas, pois fornece um conjunto vasto de exemplos de uso dos Statecharts na descrição de comportamentos híbridos. Pode constatar-se que a forma visual de explicitação do comportamento de modelos semi-empíricos revela, de um modo mais objectivo, a interligação existente entre comportamentos contínuos e discretos.

Foi, pela primeira vez, proposta uma biblioteca de modelos em Modelica para a implementação de uma parte substancial do formalismo dos Statecharts permitindo, dessa forma, a descrição de comportamentos de sistemas híbridos através de um formalismo gráfico expressivo, que pode ser facilmente compilado em Modelica. O mecanismo de modelação baseada em equações, inerente à linguagem Modelica, em conjunto com os mecanismos de ligação entre instâncias dos modelos fornecidos, revelou-se eficiente na passagem de eventos de activação e desactivação entre estados ligados hierarquicamente. Também os mecanismos de comunicação *broadcast*, do formalismo dos Statecharts, são facilmente conseguidos através da mera actualização de variáveis, sendo estas

automaticamente detectadas em todos os estados do statechart, pois, na realidade, o statechart é implementado por um sistema de equações diferenciais e algébricas que está a ser avaliado simultaneamente.

As linhas de rumo no desenvolvimento de modelos para simulação em tempo real foram a redução da rigidez e da complexidade destes. A redução da rigidez dos modelos consistiu na abstracção de algumas dinâmicas rápidas (por exemplo, o impacto do êmbolo nos extremos de um cilindro) como variações instantâneas. Este tipo de opção dá origem a modelos com múltiplos modos contínuos, intercalados com transições instantâneas, as quais são representadas com sucesso pelo formalismo dos Statecharts híbridos. A redução da complexidade dos modelos concretizou-se pela procura de modelos semi-empíricos, de modo a caracterizar alguns dos comportamentos estáticos e/ou dinâmicos dos componentes envolvidos nos sistemas hidráulicos. Para validar esta metodologia, apresentou-se um conjunto de modelos semi-empíricos e híbridos para a modelação de elementos essenciais na simulação de sistemas hidráulicos, como são as válvulas proporcionais, o óleo hidráulico ou o fenómeno do atrito em actuadores. Neste contexto, apostou-se na utilização de modelos híbridos e na modelação semi-empírica como meios de reduzir a complexidade, mantendo um bom desempenho e robustez nos modelos, com o objectivo de suportar simulações em tempo real.

Nos modelos propostos para as válvulas proporcionais de elevado desempenho (adequadas a aplicações em malha fechada) são considerados, em conjunto, os comportamentos de todos os elementos da válvula (válvula, transdutor de posição da gaveta e carta de controlo da posição da gaveta). Para o efeito foram usados dois sub-modelos desacoplados. O modelo estático reproduz os caudais volumétricos através da válvula como funções da posição da gaveta e das quedas de pressão; os modelos com melhor desempenho assumiram caudais sempre turbulentos, sendo os caudais laminares modelados implicitamente através de pseudo-secções; seguiram-se duas linhas para as funções pseudo-secção: equações com estrutura variável, baseadas na função exponencial, e equações com estrutura fixa, baseadas na função hiperbólica. O modelo dinâmico descreve o deslocamento da gaveta resultante da aplicação de um sinal eléctrico de entrada na carta de controlo; os modelos que apresentaram melhores resultados são os baseados em modelos de 2ª ordem com limitação de velocidade e aceleração da gaveta. Os modelos semi-empíricos propostos possuem algumas vantagens, relativamente aos modelos físicos que podem ser encontrados na literatura, nomeadamente a sua facilidade de parametrização e os menores custos computacionais. Os modelos estáticos desenvolvidos podem ser parametrizados directamente usando informação fornecida nos manuais dos fabricantes ou através de dados obtidos experimentalmente. Os parâmetros para os modelos dinâmicos são determinados através de técnicas de optimização bem conhecidas, usando os diagramas de Bode fornecidos pelos fabricantes. Foram apresentados

exemplos de parametrização para uma válvula da Vickers e para uma válvula da Bosch com resultados idênticos em ambos os sub-modelos estáticos e dinâmicos. Alguns modelos, nomeadamente os que assumiram caudais sempre turbulentos, reproduzem com precisão o ganho de pressão, o ganho de caudal, o caudal de fugas e a amplitude no diagrama de Bode até aos 200Hz. Quanto à fase da resposta preconiza-se a sua melhoria, especialmente para as altas frequências.

No âmbito da biblioteca de modelos, são também propostos e implementados vários modelos com diferentes complexidades para a modelação do óleo hidráulico e para as forças de atrito em actuadores. Nos modelos do óleo são evidenciadas as características mais determinantes no desempenho dinâmico de actuadores hidráulicos, como são os casos do módulo de compressibilidade e da viscosidade; os modelos considerados contemplam a variação destas características com a temperatura e com a pressão. Para a modelação do atrito consideram-se duas perspectivas diferentes: modelos de atrito estáticos e modelos dinâmicos. Os modelos de atrito dinâmicos têm vindo a ser usados nos últimos anos para a análise de servomecanismos de elevada precisão e no projecto de compensadores de atrito; estes modelos, além dos efeitos de Stribeck (típicos dos modelos estáticos clássicos), acrescentam a capacidade de representação dos deslocamentos na colagem, dos fenómenos de cola-desliza (*stick-slip*) e dos atrasos entre a variação de velocidade e a correspondente variação nas forças de atrito (*frictional lag*). O formalismo dos Statecharts é usado na descrição de vários modelos de atrito, permitindo evidenciar, durante a simulação, os estados de colagem ou deslizamento.

Um servomecanismo hidráulico, composto por uma servo-válvula proporcional e um cilindro hidráulico, foi completamente parametrizado. Foram propostos métodos para medição da frequência natural e amortecimento do conjunto, métodos para a medição das fugas internas do cilindro e métodos para a parametrização de um dos modelos de atrito dinâmico (modelo de LuGre). Na determinação dos parâmetros dinâmicos do modelo de atrito foram usadas experiências em malha aberta no servomecanismo real e num modelo não linear do sistema, sendo os parâmetros obtidos através de técnicas de optimização.

Para a realização de experiências com *hardware-in-the-loop*, foi projectada e desenvolvida uma plataforma experimental construída por um manipulador do tipo SCARA de actuação hidráulica, com hardware de aquisição de dados e de simulação e controlo em tempo real, e por um conjunto de aplicações de software de apoio. A plataforma experimental tem vindo a ser usada para a parametrização de modelos, para a avaliação do desempenho de diferentes algoritmos de controlo e, mais recentemente, na execução de testes de fadiga de componentes.



Foram também simulados sistemas hidráulicos, com *hardware-in-the-loop*, em diferentes plataformas de baixo custo. Para a simulação em tempo real recorreu-se a métodos de integração explícitos e mistos (implícitos e explícitos). A utilização dos métodos de integração mistos permitiu passos de integração maiores e mais robustez nas experiências. As pressões nas câmaras dos actuadores foram consideradas estados rápidos e integradas através do método implícito de Euler, conduzindo aos resultados esperados, isto é, a desempenhos acrescidos, já que as pressões, em termos numéricos, são os estados com variações mais bruscas. Cabe aqui uma rápida referência à utilização de controladores simples baseados em hardware (proporcionais ou proporcionais + derivativos), muito embora a plataforma construída permita o teste e parametrização de controladores mais sofisticados. Algumas experiências de simulação com *hardware-in-the-loop* foram comparadas com experiências reais equivalentes com resultados muito satisfatórios, tendo em conta que se usaram modelos semi-empíricos.

Este trabalho também mostra, incidentalmente, que a simulação com *hardware-in-the-loop* é, hoje, perfeitamente realizável em hardware de baixo custo como, por exemplo, computadores pessoais equipados com cartas de aquisição de dados, desde que se utilizem os modelos e as ferramentas de simulação adequadas.

## **Trabalho futuro**

Um trabalho desta índole levanta sempre algumas questões interessantes que podem contribuir para estender o trabalho realizado. Neste âmbito, apresentam-se alguns tópicos que poderão prosseguir o trabalho apresentado nesta dissertação:

- Actualizar a biblioteca dos Statecharts em Modelica, para a implementação de outras potencialidades dos Statecharts, tais como as transições entre níveis hierárquicos.
- Introduzir na linguagem Modelica o conceito de polimorfismo, típico das linguagens de programação por objectos, e que poderá fornecer um meio elegante e poderoso de redefinição de equações (importante na redefinição de comportamentos em modelos).
- Avaliar outras possibilidades para aproximação da fase da resposta, nos modelos dinâmicos das válvulas proporcionais.
- Utilizar outros métodos de optimização, que detectem mínimos globais, para a identificação de parâmetros, nomeadamente nos modelos dinâmicos das válvulas.

- Desenvolver modelos semi-empíricos para dois tipos de componentes importantes nos sistemas hidráulicos, e que não foram contemplados neste trabalho, como são as bombas hidráulicas e as linhas de transmissão.
- Explorar o desempenho dos modelos desenvolvidos em aplicações de projecto e teste de controladores para sistemas de actuação hidráulica; isto pressupõe a criação de uma base de dados de componentes mecânicos (nomeadamente de vedantes) onde seja possível obter parâmetros para os diferentes modelos.
- Estudar a influência de algumas das características dos dispositivos físicos de interface, necessários à implementação da simulação com *hardware-in-the-loop* (por exemplo, as frequências de amostragem dos sistemas de aquisição), nos resultados de experiências HILS.

## 8 REFERÊNCIAS

- [Almeida93] - **Gomes de Almeida, F.** Model Reference Adaptive Control of a two Axes Hydraulic Manipulator. PhD Thesis, University of Bath, UK, 1993.
- [Alur95] - **Alur, R., Courcoubetis, C., Halbwachs, N., Henzinger, T., Ho, P., Nicollin, X., Olivero, A., Sifakis, J. and Yovine, S.** The Algorithmic Analysis of Hybrid Systems. Theoretical Computer Science, 138, pp. 3-34, 1995
- [Alur97] - **Alur, R., Henzinger, T., and Wong-Toi, H.** Symbolic analysis of hybrid systems. In Proc. 37-th IEEE Conference on Decision and Control, 1997.
- [Anderson94] - **Anderson, M.** Object Oriented Modeling and Simulation of Hybrid Systems. PhD Thesis, Lund Institute of Technology, Sweden, 1994.
- [Armstrong91] - **Armstrong-Hélouvry, B.** Control of Machines with friction. Kluwer Academic Publishers, Norwell, MA, 1991
- [Armstrong94] - **Armstrong, B., Dupont, P. and Canudas de Wit, C.** A survey of models, analysis tools and compensation methods for the control of machines with friction. Automatica, 30(7), pp. 1083-1138, 1994.
- [Ascher98] - **Ascher, M. and Petzold, L.** Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations, SIAM, Philadelphia, PA, 1998.
- [Ashby56] - **Ashby, W.** An Introduction to Cybernetics, John Wiley, New York, 1956.
- [Åström98] - **Åström, K. J., Elmqvist, H. and Mattsson, S.E.** Evolution of Continuous-Time Modeling and Simulation. The 12th European Simulation Multiconference, ESM'98, Manchester, UK, June 16-19, 1998.
- [Barton94] - **Barton, P. and Pantelides, C.** Modeling of combined discrete/continuous processes. AIChE Journal, 40, pp. 966-979, 1994.
- [Barton96] - **Barton, P. and Park, T.** State Event Location in Differential-Algebraic Models. ACM Transactions on Modelling and Computer Simulation, 6(2), pp. 137-165, 1996.

- [Barton01] - **Barton, P.** and **Lee, C.** Modeling, Simulation, Sensitivity Analysis and Optimization of Hybrid Systems. In Press: ACM Transactions on Modeling and Computer Simulation: Special Issue on Multi-Paradigm Modeling, 2002.
- [Beater98] – **Beater, P.** Object Oriented Modeling and Simulation of Hydraulic Drives. In Simulation News Europe, March, 1998.
- [Beater01] - **Beater, P.** Library of Hydraulic Components for Use with Dymola. Dynasim, Sweden, 2001.
- [Beeck93] - **Beeck, M.** Integration of Structured Analysis and Timed Statecharts for real time and Concurrency Specification. In Proceedings of European Software Engineering Conference (ESEC'93), Lecture Notes in computer Science, Springer, 717, pp. 313-328, 1993.
- [Beeck94] – **Beeck, M.** A Comparison of Statecharts Variants. Lecture Notes in Computer Science 863, Springer, Berlin, 1994.
- [Beek95] - **Beek, D., Rooda, J. E.** and **Gordijn, S.** A Combined Continuous-Time / Discrete Event Approach to Modelling and Simulation of Manufacturing Machines. In Proceedings of the 1995 EUROSIM Conference, Vienna, pp. 1029-1034, 1995.
- [Beek00] – **Beek, D.** and **Rooda, J. E.** Languages and applications in hybrid modelling and simulation: Positioning of Chi. Control Engineering Practice, 8(1), pp. 81-91, 2000.
- [Benveniste90] – **Benveniste, A.** and **Le Guernic, P.** Hybrid Dynamical Systems Theory and the SIGNAL Language. IEEE Trans. on Automatic Control, 35(5), pp. 525-546, 1990.
- [Berry92] - **Berry, G.** and **Gonthier, G.** The Esterel Synchronous Programming Language: Design, Semantics, Implementation. Science of Computer Programming 19(2), pp 87-152, 1992.
- [Betts98] - **Betts, J.** Survey of Numerical Methods for Trajectory Optimization. Journal of Guidance, Control, and Dynamics. 21(2), pp. 193-207, 1998.
- [Blackburn60] – **Blackburn, J., Reethof, G.** and **Shearer, J.** Fluid Power Control. New York: Technology Press of MIT and John Wiley, 1960.
- [Bliman91] - **Bliman, P.** and **Sorine, M.** Friction modeling by hysteresis operators. Application to Dahl, sticktion, and Stribeck effects. In Proceedings of the Conference “Models of Hysteresis”, Trento, Italy, 1991.
- [Bliman95] - **Bliman, P.** and **Sorine, M.** Dry friction models for automatic control. In Proceedings of the 3rd European Control Conference, Rome, 1995.

- [Bobrow96] - **Bobrow, J. Lum, K.** Adaptive High Bandwidth Control of a Hydraulic Actuator. ASME Journal of Dynamic Systems, Measurement and Control, 118(4), 1996.
- [Borghi00] - **Borghi, M., Milani, M., and Paoluzzi, R.** Stationary axial flow force analysis on compensated spool valves. International Journal of Fluid Power, 1(1), pp. 17-25, 2000.
- [Brach91] - **Brach, R.** Mechanical Impact Dynamics: Rigid Body Collisions. John Wiley and Sons, New York, 1991.
- [Branicky95] - **Branicky, M.** Studies in Hybrid Systems: Modeling, Analysis, and Control, PhD Thesis, Massachusetts Institute of Technology, 1995.
- [Branicky98] - **Branicky, M., Borkar, V. and Mitter, S.** A unified framework for hybrid control: Model and optimal control theory. IEEE Trans. Automatic Control, 43(1), pp. 31-45, 1998.
- [Braun83] - **Braun, M., Coleman, C. and Drew, D.** Differential Equation Models. Springer-Verlag, NY, 1983.
- [Brenan88] - **Brenan, K. and Petzold, L.** The numerical solution of higher index differential/algebraic equations. Mathematics of Computation, 51, pp. 659-676, 1988.
- [Brennan98] - **Brennan, S., DePoorter, M., and Alleyne, A.** The Illinois Roadway Simulator - A Hardware-in-the-Loop Testbed for Vehicle Dynamics and Control. In proc. Of the American Control Conference, Philadelphia, Pa., pp 493-497, 1998.
- [Broenink97] - **Broenink, J.** Bond-graph modeling in Modelica. ESS'97 European Simulation Symposium, Passau, Germany, 1997.
- [Brogliato96] - **Brogliato, B.** Nonsmooth Impact Mechanics: Models, Dynamics and Control. Springer, London, UK, 1996.
- [Brun99] - **Brun, X., Belgharbi, M., Sesmat, S., Thomasset, D. and Scavarda, S.** Control of an electropneumatic actuator: comparison between some linear and non-linear control laws. Proc. Instn. Mech. Engrs, Part I, Journal of Systems and Control Engineering, 213, pp. 387-406, 1999.
- [Buisson97] - **Buisson, J. and Cormerais, H.** Modeling Hybrid Linear Systems With Bond-Graph Using an Implicit Formulation, Bond Graph Digest, 1(1), 1997.
- [Bullo98] - **Bullo, F. and Zefran, M.** On modeling and locomotion of hybrid mechanical systems with impacts. IEEE Conference on Decision and Control, 1998.
- [Burrows72] - **Burrows, C.** Fluid Power Servomechanisms, Van Nostrand Reinhold Company, London, England, 1972.

- [Burrows98] - **Burrows, C.** Fluid Power Systems – An Academic Perspective. JHPS Journal of Fluid Power Systems, 29(1), pp. 26-32, 1998.
- [Bush31] – **Bush, V.** The differential Analyzer: A new machine for solving differential equations. Journal of the Franklin Institute, 212, pp. 447-488, 1931.
- [Buzdugan99] - **Buzdugan, L., Balling, O., Lee, P., Balling, C., Freeman, J. and Huck, F.** Multirate Integration for Real-Time Simulation of Wheel Loader Hydraulics. In Proceedings of DETC'99 ASME Design Engineering Technical Conferences, September, Las Vegas, Nevada, USA, 1999.
- [Byrne92] - **Byrne, G. and Schiesser, W.** Recent Developments in Numerical Methods and Software for ODEs/DAEs/PDEs, World Scientific, Singapore, 1992.
- [Canudas95] - **Canudas de Wit, C., Olsson, H., Aström, K. and Lischinsky, P.** A new model for Control of Systems with Friction. IEEE Transactions on Automatic Control, 40(3), 1995.
- [Canudas97] - **Canudas de Wit, C. and Lischinsky, P.** Adaptive Friction Compensation with Partially Known Dynamic Friction Model. International Journal of Adaptive Control and Signal Processing, 11(1), pp. 65-80, 1997.
- [Cassandras97] - **Cassandras, C. and Pepyne, D.** Optimal control of a class of hybrid systems. In Proceedings of the IEEE Conference on Decision and Control, San Diego, California, December, pp 133-138, 1997.
- [Cellier91] - **Cellier, F.** Continuous System Modeling, Springer-Verlag, New York, 1991.
- [Cellier96] - **Cellier, F. Elmqvist, H. and Otter, M.** Modeling from Physical Principles. In The Control Handbook, ed W. S. Levine, CRC Press, Boca Raton, FL, pp 99-107, 1996.
- [Champagnat98] - **Champagnat, R., Esteban, P., Pingaud, H. and Valette, R.** Petri net based modeling of hybrid systems. Computers in industry, 36(1-2), pp. 139-146, 1998.
- [Chatterjee98] - **Chatterjee, A.** A simple two-parameter collision model for general 3D frictional, single-point rigid body collisions. Journal of Applied Mechanics, 65(4), pp. 939-951, 1998.
- [Coleman92] - **Coleman, D., Hayes, F. and Bear, S.** Introducing Objectcharts or How to Use Statecharts in Object-Oriented Design. IEEE Transactions on Software Engineering, 18(1), 1992.
- [Coleman99] - **Coleman, T., Branch, A. and Grace, A.** Optimization Toolbox, The MathWorks Inc., 1999.
- [Dahl76] - **Dahl, P.** Solid friction damping of mechanical vibration. AIAA Journal, 14(12), pp.1675-1682, 1976.

- [David92] - **David, R.** and **Alla, H.** Petri Nets & GRAFCET: Tools for Modelling Discrete Event Systems, Prentice-Hall, 1992.
- [Deshpande98] - **Deshpande, A., Göllü, A.** and **Semenzato, L.** The SHIFT programming language for dynamic networks of hybrid automata. IEEE transactions on Automatic Control, 43(4), pp. 584-587, 1998.
- [DiMaio98] - **DiMaio, S., Salcudean, S., Reboulet, C., Tafazoli, S.** and **Hashtrudi-Zaad, K.** A Virtual Excavator for Controller Development and Evaluation. In Proceedings of the 1998 IEEE International Conference on Robotics & Automation, Belgium, 1998.
- [Dreyer93] - **Dreyer, T.** Modeling with Ordinary Differential Equations. CRC, Boca Raton, 1993.
- [Dupond00] - **Dupont, P., Armstrong, B., Hayward, V.** and **Altpeter, V.** Elasto-Plastic Friction Models: Contact Compliance and Stiction. In Proceedings of the 2000 American Control Conference, June, Chicago, III. USA, 2000.
- [Dupond02]- **Dupont, P., Hayward, V., Armstrong, B.** and **Altpeter, V.** Single State Elasto-Plastic Friction Models. IEEE Transactions on Automatic Control, 47(5), pp.787-792, 2002.
- [Durinsky89] - **Durinsky, D.** and **Harel, D.** Using Statecharts for Hardware Description and Synthesis. IEEE Trans. Computer-Aided Design, 8, pp.798-807, 1989.
- [Dusinsky97] - **Dusinsky, D.** BetterState Pro Tutorial: An Introduction to Designing with Statecharts. Integrated Systems, Inc. ([www.isi.com](http://www.isi.com)), 1997.
- [Edge97] - **Edge, K.** The control of fluid power systems – responding to the challenges. Proc. Instn. Mech. Engrs, Part I, Journal of Systems and Control Engineering, 211, pp. 91-110, 1997.
- [Ellman95] - **Ellman, A., Sanerma, S., Salminen, M., Piché, R.** and **Virvalo, T.** Tools for Control and Hydraulic Circuit Design of a Hydraulic-Driven Manipulator Mechanism. In Proceedings of the 9th European Simulation Multiconference, June, Czech Republic, 1995.
- [Ellman98] - **Ellman, A.** Leakage behaviour of four-way servovalve. In Fluid Power Systems and Technology 1998, FPST Vol 5, Collected papers of 1998 ASME IMECE, Anaheim, November, pp. 163-167, 1998.
- [Elmer99] - **Elmer, K.** Mathematical models for a range of electrohydraulic proportional control valves. MPhil Thesis, The Nottingham Trent University, UK, 1999.
- [Elmqvist78] – **Elmqvist, H.** A structured Model Language for Large Continuous Systems. PhD Thesis, Dep. of Automatic Control, Lund Institute of Technology, Sweden, 1978.

- [Elmqvist94] - **Elmqvist, H.** and **Otter, M.** Methods for Tearing Systems of Equations in Object-Oriented Modelling. In Proceedings ESM'94 European Simulation Multiconference, Barcelona, Spain, pp. 326-332, 1994.
- [Elmqvist95] - **Elmqvist, H., Otter, M.** and **Cellier, F.** Inline Integration: A New Mixed Symbolic/Numeric Approach for Solving Differential-Algebraic Equation Systems, Proc. ESM'95, SCS European Simulation MultiConference, Prague, Czech Republic, 1995.
- [Elmqvist97] - **Elmqvist, H.** and **Mattsson, S. E.** Modelica - The Next Generation Modeling Language - An International Design Effort. First World Congress of System Simulation, Singapore, 1997.
- [Elmqvist00] - **Elmqvist, H., Mattsson, S. E.** and **Otter, M.** Object-Oriented and Hybrid Modeling in Modelica. ADPM 2000, Dortmund, Germany, 2000.
- [Elmqvist02] - **Elmqvist, H., Mattsson, S. E.,** and **Olsson, H.** New Methods for Hardware-in-the-Loop Simulation of Stiff Models. In Proceedings of the 2nd International Modelica Conference, March, Oberpfaffenhofen, Germany, 2002.
- [Engelbrecht99] - **Engelbrecht, R. Poe, C.** and **Balke, K.** Development of a Distributed Hardware-In-The-Loop Simulation System for Transportation Networks. In Proceedings of the 78th Annual Meeting of the Transportation Research Board, 1999.
- [Fábian99] - **Fábian, G.** A Language and Simulator for Hybrid Systems, PhD Thesis, Eindhoven University of Technology, Netherlands, 1999.
- [Fábian00] - **Fábian, G., Beek, D.** and **Rooda, J.** Substitute equations for index reduction and discontinuity handling. In Proceedings of the Third International Symposium on Mathematical Modelling, Vienna, 2000.
- [Fahrland70] - **Fahrland, D.** Combined discrete event continuous system simulation. Simulation 14(2), pp. 61-72, 1970.
- [Feky99] - **Feki, M., Richard, E.** and **Gomes Almeida, F.** Commande en effort d'un vérin hydraulique par linéarisation entrée/sortie. In Proceedings of Journées Doctorales d'Automatique, JDA'99, Nancy, France, pp 181-184, 1999.
- [Ferreira99a] - **Ferreira, J. A.** and **Estima de Oliveira, J. P.** Modeling Hybrid Systems Using Statecharts and Modelica. 7th IEEE International Conference on Emerging Technologies and Factory Automation, Barcelona, Spain, 1999.



- [Ferreira99b] - **Ferreira, J. A., Estima de Oliveira, J. P., and Costa, V. A.** Modeling of Hydraulic Systems for Hardware-in-the-loop Simulation: a Methodology Proposal. In Proceedings of the 1999 International Mechanical Engineering Congress & Exposition Nashville, USA, 1999.
- [Ferreira02] - **Ferreira, J. A., Gomes de Almeida, F. and Quintas, M. R.** Semi-empirical model for a hydraulic servo-solenoid valve. Proc. Instn Mech. Engrs, Part I, Journal of Systems and Control Engineering, Vol 216, pp. 237-248, 2002.
- [Fonseca00] - **Fonseca, F. e Louro, S.** Controlo de um Manipulador do Tipo SCARA. Projecto Final da Licenciatura em Eng. Electrónica e Telecomunicações, Universidade de Aveiro, 2000.
- [Gafvert96] - **Gafvert, M.** Comparasion of two Friction Models. Master Thesis, Lund Institute of Technology, University of Lund, 1996.
- [Gawthrop00] - **Gawthrop, P.** Symbolic Generation of Real-Time Simulation Code for Large Stiff Nonlinear Systems. In Proceedings of the UKACC International Conference CONTROL 2000 University of Cambridge, UK, 2000.
- [Gear71] - **Gear, C.** Numerical Initial Value Problems in Ordinary Differential Equations, Automatic Computation. Prentice-Hall, Englewood Cliffs, New Jersey, 1971.
- [Gear84] - **Gear, C. and Petzold, L.** ODE Methods for the Solution of Differential/Algebraic Systems, SIAM Journal on Numerical Analysis, 21, pp. 367-384, 1984.
- [Gear88] - **Gear, C.** Differential-algebraic equation index transformations. SIAM. J. Sci. Stat. Comp., 9, pp. 39-47, 1988.
- [Geuder95] - **Geuder, D.** Object-Oriented Modeling with SIMPLE ++. In proceedings of the 1995 Winter Simulation Conference, Virginia, USA, pp. 534-540, 1995.
- [Gonthier98] - **Gonthier, Y. and Papadopoulos, E.** On the Development of a Real-time Simulator for an Electro-hydraulic Forestry Machine. In Proceedings of the 1998 IEEE International Conference on Robotics & Automation, Leuven, Belgium, pp. 127-132, 1998.
- [Haessig91] - **Haessig, D. and Friedland, B.** On the modelling and simulation of friction. ASME Journal of Dynamic Systems, Measurement and Control, 113, pp. 354-362, 1991.
- [Halbwachs91] - **Halbwachs, N., Caspi, P., Raymond, P. and Pilaud, D.** The synchronous dataflow programming language Lustre. In Proceedings of the IEEE, 79(9), 1991.
- [Halbwachs93] - **Halbwachs, N.** Synchronous Programming of Reactive Systems. Kluwer, 1993.
- [Handroos91] - **Handroos, H. and Vilenius, M.** Flexible Semi-Empirical Models for Hydraulic Flow Control Valves. Journal of Mechanical Design, 113(3), pp. 232-238, 1991.

- [Hanselmann96] - **Hanselmann, H.** Hardware-in-the-Loop Simulation Testing and its Integration into a CACSD Toolset. In Proceedings of the IEEE International Symposium on Computer-Aided Control System Design September, Dearborn, Michigan, USA, 1996.
- [Harel87] - **Harel, D.** Statecharts: A visual formalism for complex systems. *Science of Computer Programming* 8, 1987.
- [Harel90] - **Harel, D., Lachover, H., Naamad, A., Pnueli, A., Politi, M., Sherman, R., Shtull-Trauring, A. and Trakhtenbrot, M.** Statemate: a working environment for the development of complex reactive systems. *IEEE Transactions on Software Engineering*, 16(4), pp. 403-414, 1990.
- [Harel92] – **Harel, D. and Kahane, C.** On Statecharts with overlapping. *ACM Transactions on Software Engineering and Methodology*, 1(4), pp. 399-421, 1992.
- [Harel96] - **Harel, D. and Naamad, A.** The STATEMATE semantics of statecharts. *ACM Transactions on Software Engineering and Methodology*, 5(4), pp. 293-333, 1996.
- [Harel97a] - **Harel, D. and Gery, E.** Executable Object Modeling with Statecharts. *IEEE Computer*, July 1997.
- [Harel97b] - **Harel, D., Pnueli, A., Schmidt, J. and Sherman, R.** On the formal semantics of statecharts. In Proceedings of the 2nd IEEE Symposium on Logic in Computer Science, IEEE Press, Ithaca, New York, pp. 54-64, 1987.
- [Hayhard70] - **Hayward, A.** How to estimate the bulk modulus of hydraulic fluids. *Journal, Institute of Petroleum*, 56, pp. 12-32, 1970.
- [Heath97] - **Heath, M.** *Scientific Computing: An Introductory Survey.* McGraw-Hill, 1997.
- [Hedenetz 98] - **Hedenetz, B. and Schedl, A.** Fault Injection and Fault Modeling for a Safety Critical Automotive Communication System. In Proceedings of the European conference on safety and reliability. ESREL '98, Trondheim (NO), 1998.
- [Hong95] - **Hong, H., Kim, J., Cha, S. and Know, Y.** Static Semantics and Priority schemes for Statecharts. In Proceedings of Computer Software and Applications Conference '95, Texas USA, 1995.
- [Hooman92] - **Hooman, J., Ramesh, S. and Roever, W.** A compositional axiomatization of Statecharts. *Theoretical Computer Science*, 101, pp. 289-335, 1992.
- [Hopcroft79] - **Hopcroft, J. and Ullman, J.** *Introduction to Automata Theory, Languages and Computation.* Addison-Wesley, 1979.

- [Hwang97] - **Hwang, S., Heo, S., Kim, H. and Lee, K.** Vehicle Dynamic Analysis and Evaluation of Continuously Controlled Semi-Active Suspensions Using Hardware-in-the-Loop Simulation. *Vehicle System Dynamics*, 27, pp. 423-434, 1997.
- [Ikenaga99] - **Ikenaga, S., Lewis, F., Davis, L., Campos, J., Evans, M. and Scully, S.** Active Suspension Control Using a Novel Strut and Active Filtered Feedback: Design and Implementation. In *The 8th IEEE International Conference on Control Applications*, Kohala Coast-Island of Hawaii HI, 1999.
- [Isermann99] - **Isermann, R., Schaffnit, J. and Sinsel, S.** Hardware-in-the-loop simulation for the design and testing of engine-control systems. *Control Engineering Practice*, 7, pp. 643 – 653, 1999.
- [Istefanian96] - **Istefanian, H., Pratt, G. and Jones, S.** Effect of Fixed-Point Parameterization on the Performance of Active Suspension Control Systems. In *Proceedings IFAC 13th. World Congress*, San Francisco, USA, pp. 291-297, 1996.
- [Jansson94] - **Jansson, A. and Palmberg, J. O.** Load simulation, a flexible tool for assessing the performance of hydraulic valves. In *Proceedings of the Fourth Triennial International Symposium on Fluid Control, Fluid Measurement, and Visualisation*, Toulouse, France, 1994.
- [Jeandel96] - **Jeandel, A., Boudaud, F., Ravier, P. and Buhsing, A.** U.L.M: Un Langage de Modélisation, a modeling language. In *Proceedings of the CESA'96 IMACS Multiconference*, Lille, France, 1996.
- [Jinghong94] - **Jinghong, Y., Zhaoneng, C. and Yuanzhang, L.** The Variation of Oil Effective Bulk Modulus With Pressure in Hydraulic Systems. *ASME Journal of Dynamic Systems, Measurement and Control*, 116, pp. 146-149, 1994.
- [Kapur97] - **Kapur, A.** Interval and Point-Based Approaches to Hybrid System Verification, PhD Thesis, Stanford University, USA, 1997.
- [Karnopp85] - **Karnopp D.** Computer simulation of stick-slip friction in mechanical dynamic systems. *ASME Journal of Dynamic Systems, Measurement and Control*, 107, pp. 100-103, 1985.
- [Katok95] - **Katok, A. and Mendoza, L.** Introduction to the Modern Theory of Dynamical Systems Cambridge University Press, 1995.
- [Kesten92] - **Kesten, Y. and Pnueli, A.** Timed and hybrid statecharts and their textual representation. In: J.Vytopil (ed.): *Formal Techniques in Real-Time and Fault-Tolerant Systems*. Springer-Verlag, 1992.

- [Khaimovich65] - **Khaimovich, E.** Hydraulic Control of Machine Tools, Pergamon Press, 1965.
- [Kirkerud89] - **Kirkerud, B.** Object-Oriented Programming with SIMULA, Addison-Wesley, 1989.
- [Kloas95] – **Kloas, M., Friesen, V. and Simons, M.** Smile A Simulation Environment for Energy Systems. In Proceedings of the 5th International IMACS-Symposium on Systems Analysis and Simulation, Sydow, A. (ed.), System Analysis Modelling Simulation series, Gordon and Breach Publishers, 18, 1995.
- [Kohavi78] - **Kohavi, Z.** Switching and Finite Automata Theory. New York: McGraw-Hill, 1978.
- [Koskinen92] – **Koskinen, K. and Virvalo, T.** Fuzzy Logic Controller for Hydraulic Drives. In Proceedings of the 10th Aachen Colloquium on Fluid Power Technology. Aachen, Germany, 1992.
- [Koskinen00] - **Koskinen, K. and Vilenius, M.** Steady State and Dynamic Characteristics of Water Hydraulic Proportional Ceramic Spool Valve. International Journal of Fluid Power, 1(1), pp. 5-15, 2000.
- [Krasnosel89] - **Krasnosel, M. and Pokrovskii, A.** Systems with hysteresis. Springer-Verlag, Berlin Heidelberg, 1989.
- [Lagarias98] - **Lagarias, J., Reeds, J., Wright, M. and Wright, P.** Convergence Properties of the Nelder-Mead Simplex Method in Low Dimensions. SIAM Journal of Optimization, 9(1), pp.112-147, 1998.
- [Lambert91] - **Lambert, J.** Computational Methods in Ordinary Differential Equations, John Wiley & Sons, 1991.
- [Lane01] - **Lane, D., Falconer, G., Randall, G., Edwards, I., Evans, J., Smith, J., Hunter, J., Standeven, J. and Colley, M.** Interoperability and Synchronisation of Distributed Hardware-in-the-loop Simulation for Underwater Robot Development: Issues and Experiments. In Proceedings of the 2001 IEEE Conference on Robotics and Automation, Seoul, Korea, pp. 909-914, 2001.
- [Lebrun87] - **Lebrun, M.** A Model for a Four-Way Spool Valve Applied to a Pressure Control System. The Journal of Fluid Control, pp. 38-54, 1987.
- [Lee99] - **Lee, S. and Kim, J.** Friction Identification Using Evolution Strategies and Robust Control of Positioning Tables. ASME Journal of Dynamic Systems, Measurement and Control, 121 pp. 619-624, 1999.

- [Lennevi95] - **Lennevi, J., Palmberg, J. and Jansson, A.** Simulation Tool for the Evaluation of Control Concepts for Vehicle Drive Systems. In Proceedings of the 4th Scandinavian International Conference on Fluid Power, Tampere, September, 1995.
- [Lin97] - **Lin, J. and Kanellakopoulos, I.** Nonlinear design of active suspensions. IEEE Control Systems, 17(3), pp. 45-59, 1997.
- [Linjama98] - **Linjama, M.** The Modeling and Actuator Space Control of Flexible Hydraulic Cranes. PhD Thesis, Institute of Hydraulics and Automation. Tampere University of Technology, Finland, 1998.
- [Lischinsky99] - **Lischinsky, P., Canudas-de-Wit, C. and Morel, G.** Friction Compensation for an Industrial Hydraulic Robot. IEEE Control Systems Magazine, 19(1), pp. 25-32, 1999.
- [Lygeros94] - **Lygeros, J., Godbole, D. and Sastry, S.** Simulation as a Tool for Hybrid system design. In Proceedings of the Fifth IEEE conference on AI, Simulation and Planning in High-Autonomy Systems, pp. 6-12, 1994.
- [Maclay97] - **Maclay, D.** Simulation gets into the loop. In IEE Review, May, pp. 109-112, 1997.
- [Mansoor00] - **Mansoor, S., Jones, D., Bradley, D., Aris, F. and Jones, G.** Hardware-In-The-Loop Simulation of a Pumped Storage Hydro Station. In Proceedings of the IASTED International Conference PES2000, Marbella, Spain, pp. 571-577, 2000.
- [Mattsson93] - **Mattsson, S. E. and Söderlind, G.** Index reduction in differential-algebraic equations. SIAM Journal of Scientific and Statistical Computing, 14(3), pp.677-692, 1993.
- [McCabe01] - **McCabe, C., Cui, S. and Cummings, P.** Characterizing the viscosity-temperature dependence of lubricants by molecular simulation. Fluid Phase Equilibria, pp. 363-370, 2000.
- [McCloy73] - **McCloy, D. and Martin, H.** The control of fluid Power. Longman Group Limited, London, 1973.
- [Merrit67] - **Merrit, H.** Hydraulic control systems. John Wiley & Sons, New York, 1967.
- [Meyer92] - **Meyer, B.** Eiffel: The Language, Prentice Hall, 1992.
- [Mitchell76] - **Mitchell, E. and Gauthier, J.** Advanced Continuous Simulation Language. Simulation, pp. 72-78, 1976.
- [Moskwa95] - **Moskwa, J., Weeks, R. and Moskwa, J.** Automotive Engine Modeling for Real-time Control Using MATLAB/SIMULINK, Society of Automotive Engineers SAE paper No. 950417, 1995.

- [Mosterman95] - **Mosterman, P.** and **Biswas, G.** Behavior Generation Using Model Switching - A Hybrid Bond Graph Modeling Technique. 1995 International Conference on Bond Graph Modeling and Simulation (ICBGM'95), LasVegas, SCS Publishing, Simulation Series, 27(1), pp. 177-182, 1995.
- [Mosterman97] – **Mosterman, P.** Hybrid Dynamic Systems: A hybrid Bond Graph Modeling Paradigm and its Application in Diagnosis. PhD Thesis, Vanderbilt University, Tennessee, USA, 1997.
- [Mosterman98] - **Mosterman, P., Otter, M.** and **Elmqvist, H.** Modeling Petri Nets as Local Constraint Equations for Hybrid Systems Using Modelica. In Proceedings of the Summer Computer Simulation Conference '98, Reno, Nevada, USA, pp. 314-319, 1998.
- [Mosterman99] - **Mosterman, P.** An Overview of Hybrid Simulation Phenomena and Their Support by Simulation Packages. In Hybrid Systems: Computation and Control '99, Lecture Notes in Computer Science, 1569, pp. 165-177, 1999.
- [Mosterman00] - **Mosterman, P.** and **Biswas, G.** A Comprehensive Methodology for Building Hybrid Models of Physical Systems. Artificial Intelligence, 121, pp. 171-209, 2000.
- [Münzer01] - **Münzer, M.** and **Pedersen, P.** Real Time Visual Simulation Of Flexible Mobile Crane For Machine-Operator Interaction Testing. In Proceedings of the 2nd International Workshop Ostrava - Malenovice, 2001.
- [Newton95] - **Newton, D.** Application of a neural network controller to control a rotary system with a high power efficiency. In Innovations in Fluid Power, Proceedings of the Seventh Bath International Fluid Power Workshop, 1995.
- [Norris95] - **Norris, G.** Boeing's seventh wonder. IEEE Spectrum, October, pp. 20-23, 1995.
- [Olsson96] - **Olsson, H.** Control Systems with Friction. PhD thesis, Lund Institute of Technology, University of Lund, 1996.
- [Olsson97] - **Olsson, H., Astrom, K., Canudas de Wit, C., Gafvert, M,** and **Lischinsky, P.** Friction Models and Friction Compensation. European Journal of Control, 4, pp. 176-195, 1998.
- [Otter95] - **Otter, M.** and **Cellier, F.** Software for Modeling and Simulating Control Systems, The Control Handbook (W.S. Levine, ed.), CRC Press, Boca Raton, FL, pp.415-428, 1995.
- [Otter97] - **Otter, M., Schlegel, C.** and **Elmqvist, H.** Modeling and Realtime Simulation of an Automatic Gearbox using Modelica. In Proceedings of the European Simulation Symposium, ESS'97, Passau, Germany, 1997.

- [Otter99] - **Otter, M., Elmqvist, H. and Mattsson, S. E.** Hybrid Modeling in Modelica based on the Synchronous Data Flow Principle. 1999 IEEE Symposium on Computer-Aided Control System Design, CACSD'99, Hawaii, pp. 151-157, 1999.
- [Otter00a] - **Otter, M., Dempsey, M. and Schlegel, C.** Package PowerTrain: A Modelica library for modeling and simulation of vehicle power trains. Modelica Workshop, Lund, Sweden, 2000.
- [Otter00b] - **Otter, M., Remelhe, M., Engell, S. and Mosterman, P.** Hybrid Models of Physical Systems and Discrete Controllers. *Automatisierungstechnik*, 48, 2000.
- [Paynter61] - **Paynter, H.** Analysis and design of Engineering Systems. MIT Press, Cambridge Mass, 1961.
- [Pedgen95] - **Pedgen, C., Shannon, R. and Sadowski, R.** Introduction to simulation using SIMAN. London McGraw-Hill, 1995.
- [Peng97] - **Peng, H., Strathearn, R. and Ulsoy, A.** A Novel Active Suspension Design Technique-Simulation and Experimental Results. In Proceedings of the 1997 American Control Conference, Albuquerque, New Mexico, 1997.
- [Petri73] - **Petri, C.** Concepts of Net Theory. *Mathematical Foundations of Computer Science: In Proceedings of the Symposium and Summer School, High Tatras*, 1973.
- [Piela91] - **Piela, P., Epperly, T., Westerberg, K. and Westerberg, A.** ASCEND: An Object Oriented Computer Environment for Modeling and Analysis. *Computers and Chemical Engineering*, 15, pp. 53-72, 1991.
- [Pnueli91] - **Pnueli, P. and Shalev, M.** What is in a Step: On the semantics of Statecharts. In: *Theoretical Aspects on Computer S, Lecture Notes in Computer Science*, Springer-Verlag, Berlin, 256, 1991.
- [Praehofer91] - **Praehofer, H.** System Theoretic Foundations for Combined Discrete-Continuous System Simulation, PhD thesis, University of Linz, Austria, 1991.
- [Quintas99] - **Quintas, M.** Contribution à la Modélisation et à la Commande Robuste des Systèmes Electrohydrauliques. Thèse de doctorat, L'Institut National des Sciences Appliquées de Lyon, France, 1999.
- [Rimvall86] - **Rimvall, M. and Cellier, F.** Evolution and perspectives of simulation languages follow the CSSL standard. *Modeling, Identification and Control*, 6, pp 181-199, 1986.
- [Rothenberg89] - **Rothenberg, J.** The nature of modeling. In L.E. Widman, K.A. Loparo and N.R. Nielsen, *Artificial intelligence, simulation & modeling*. New York: Wiley, 1989.

- [Sahlin96] – **Sahlin, P., Bring A. and Sowell E.** The Neutral Model Format for Building Simulation, Version 3.02. Department of Building Sciences, The Royal Institute of Technology, Stockholm, Sweden, 1996.
- [Sastry95] - **Sastry, S., Meyer, G., Tomlin, C., Lygeros, J., Godbole, D. and Pappas, G.** Hybrid systems in air traffic control. In Proceedings of the IEEE Conference on Decision and Control, pp. 1478-1483, 1995.
- [Schaft98] - **Schaft, J. and Schumacher, H.** Complementary modelling of hybrid systems. IEEE Transactions on Automatic Control, 43(4), pp. 483-490, 1998.
- [Schiela00] - **Schiela, A. and Olsson, H.** Mixed-mode Integration for Real-Time Simulation. In Proceedings of the First International Modelica Workshop, Lund, Sweden, pp. 69-75, 2000.
- [Scheila02] - **Schiela, A. and Bornemann, F.** Sparsing in Real Time Simulation. Accepted for publication in ZAMM Z. Math. Mech., 2002.
- [Schlegel02] - **Schlegel C., Bross M. and Beater P.** HIL Simulation of the Hydraulics and Mechanics of an Automatic Gearbox. In Proceedings of the 2nd International Modelica Conference, Oberpfaffenhofen, Germany, 2002.
- [Schmidt00] - **Schmidt, A., Gold, P., Abmann, C., Dicke, H. and Loos, J.** Viscosity-Pressure-Temperature Behaviour of Mineral and Synthetic Oils. In Proceedings of the 12th International Colloquium Tribology 2000 - Plus, Stuttgart/Ostfildern, Germany, 2000.
- [Schmidt87] - **Schmidt, B.** Model Construction with GPSS-Fortran Version 3. Springer Verlag, Berlin, 1987.
- [Schothorst97] - **Schothorst, G.** Modelling of Long-Stroke Hydraulic Servo-Systems for Flight Simulator Motion Control and System Design. PhD Thesis, Mechanical Engineering Systems and Control Group, Delft University of Technology, The Netherlands, 1997.
- [Selic94] - **Selic, B., Gullekson, G. and Ward, P.** Real Time Object Oriented Modeling, John Wiley Sons Inc, 1994.
- [Shampine94] - **Shampine, L.** Numerical solution of ordinary differential equations, Chapman & Hall, 1994.
- [Shetty01] - **Shetty, D., Kolk, R., Kondo, J. and Campana, C.** A New Approach to Mechatronics Systems Design Using Hardware in the Loop Simulation. In Proceedings of the IEEE/ASME Intl. Conf. On Advanced Intelligent Mechatronics (AIM'01), Como, Italy, pp. 1005-1010, 2001.



- [Silva01] – **Silva, C.** Simulação Digital de Sistemas Óleo-Hidráulicos em Computadores Pessoais. Dissertação para acesso à categoria de Investigador Auxiliar, Universidade do Porto, 2001.
- [Sinha01] - **Sinha, R., Liang, V., Paredis, C. and Khosla, P.** Modeling and Simulation Methods for Design of Engineering Systems. Journal of Computing and Information Science in Engineering, ASME, 1(1), pp. 84-91, 2001.
- [Söderman95] - **Söderman, U. and Strömberg, J.** Switched bond graphs: Towards systematic composition of computational models. In ICBGM95, Las Vegas, pp. 73-79, 1995.
- [Soejima02] - **Soejima S. and Matsuba, T.** Application of mixed mode integration and new implicit inline integration at Toyota. In Proceedings of the 2nd International Modelica Conference, 2002, Oberpfaffenhofen, Germany, 2002.
- [Stolpe98] - **Stolpe, R. and Zanella, M.** A Distributed Hardware-in-the-Loop Simulation Environment in Use on a Testbed of a Series Hybrid Drive. In Proceedings of the ESM98 - 12th European Simulation Multiconference, Manchester, UK, 1998.
- [Strauss67] - **Strauss, J.** The SCI Continuous Simulation Language (CSSL). Simulation, 9, pp. 281-308, 1967.
- [Stroustrup85] - **Stroustrup, B.** The C++ Programming Language, Addison-Wesley, 1985.
- [Suh00] - **Suh, M., Chung, J., Seok, C. and Kim, Y.** Hardware-In-The-Loop Simulation for ABS based on PC, International Journal of Vehicle Design, 24, pp. 157-170, 2000.
- [Sweet95] - **Sweet, W.** The glass cockpit, IEEE Spectrum, September, pp. 30-38, 1995.
- [Talaia02] - **Talaia, P., Rocha, A. and Ferreira, J. A.** Suspension Analysing Equipment Based on a Hydraulic Servo-Cylinder. In Proceedings of CONTROLO 2002 - 5th Portuguese Conference on Automatic Control University of Aveiro, 2002.
- [Teukolsky92] - **Teukolsky, S., Vetterling, W. and Flannery, B.** Numerical Recipes in C, The Art of Scientific Computing, Cambridge University Press, 1992.
- [Tittus95] - **Tittus, M. and Egardt, B.** Hybrid objects. In Hybrid Systems II, Lect. N. in Comp. Sci. 999, Springer Verlag, pp. 493-508, 1995.
- [Tomlin98] - **Tomlin, C., Lygeros, J. and Sastry, S.** Synthesizing controllers for nonlinear hybrid systems. In Hybrid Systems: Computation and Control, in LNCS, New York: Springer Verlag, 1386, pp. 360-373, 1998.

- [Turine98] - **Turine, H.** Um Modelo Baseado em Statecharts para a Especificação Formal de Hiperdocumentos. Tese de Doutorado, Instituto de Física de São Carlos (IFSC), São Carlos, SP, 1998.
- [Vaughan96] - **Vaughan, N.** and **Gamble, J.** The Modelling and Simulation of a Proportional Solenoid Valve. ASME Journal of Dynamic Systems, Measurement and Control, 118 (1), pp. 120-125, 1996.
- [Viersma80] - **Viersma, T.** Analysis, Synthesis and Design of Hydraulic Servosystems and Pipelines. Elsevier Scientific Publishing Company, Amsterdam, The Netherlands, 1980.
- [Viklund95] - **Viklund, L.** and **Fritzson, P.** ObjectMath an Object-Oriented Language and Environment for Symbolic and Numerical Processing in Scientific Computing. Scientific Programming, 4, pp. 229-250, 1995.
- [Virtalo97] - **Virtalo, T.** Nonlinear model of analog valve. In the 5th Scandinavian International Conference on Fluid Power, Linköping, May, pp. 199-214, 1997.
- [Watton89] - **Watton, J.** Fluid Power Systems – Modeling, simulation, analog and microcomputer control, Prentice Hall, UK, 1989.
- [Weber96] - **Weber, M.** Combining Statecharts and Z for the Design of Safety-Critical Control Systems. Proc. Int. Conf. FME 96; Industrial Benefit and Advances in Formal Methods, 1996.
- [Wong-Toi97] - **Wong-Toi, H.** The synthesis of controllers for linear hybrid automata. In Proceedings of the 36th IEEE CDC, San Diego, CA, 1997.
- [Zeigler76] - **Zeigler, B.** Theory of Modeling and Simulation, John Wiley, New York, 1976.
- [Zlajpah98] - **Zlajpah, L.** Hardware-In-The-Loop Simulation Using Planar Manipulators Toolbox. In Proceedings of EUROSIM Simulation Congress, Helsinki, Finland, pp. 565-572, 1998.
- [ModSpecific00] - **Modelica Association** ModelicaTM - A Unified Object-Oriented Language for Physical Systems Modeling. Specification, Version 1.4, 2000.
- [ModTutorial00] - **Modelica Association** ModelicaTM - A Unified Object-Oriented Language for Physical Systems Modeling. Tutorial, Version 1.4, 2000.
- [dSPACE] - dSPACE gmbH, DS1102 DSP Controller Board, Technical Manuals, Technologiepark, Paderborn, Germany.
- [Stateflow97] - StateFlow - The MATH WORKS Inc., Natick, USA. ([www.mathworks.com](http://www.mathworks.com)).
- [Vickers99] - Vickers/Eaton Aerospace Hydraulic Division. Solenoid Operated Proportional Valves, Technical Data Sheet, Mississippi, USA