



Universidade do Minho  
Escola de Engenharia

Nuno Miguel Evangelista Canadas

Modelação da parte física de sistemas mecatrónicos e estudo da sua influência em simulação MiL (*Model-in-the-loop*)





Universidade do Minho  
Escola de Engenharia

Nuno Miguel Evangelista Canadas

Modelação da parte física de sistemas  
meatrónicos e estudo da sua influência  
em simulação MiL (*Model-in-the-loop*)

Tese de Mestrado  
Ciclo de Estudos Integrados Conducentes ao  
Grau de Mestre em Engenharia Mecatrónica

Trabalho efetuado sob a orientação do  
Professor Doutor José Mendes Machado

## DECLARAÇÃO

Nome: Nuno Miguel Evangelista Canadas

Correio eletrónico: nuno\_canadas@hotmail.com

Tel./Tlm.: 914688091

Número do Bilhete de Identidade: 13757810

Título da dissertação:

Modelação da parte física de sistemas mecatrónicos e estudo da sua influência em simulação MiL (*Model-in-the-loop*)

Ano de conclusão: 2013

Orientador(es): Professor Doutor José Mendes Machado

Designação do Mestrado: Mestrado em Engenharia Mecatrónica

Ciclo de Estudos Conducentes ao Grau de Mestre em Engenharia

Área de Especialização: Automação

Escola: Escola de Engenharia

Departamento: Departamento de Engenharia Mecânica

1. É AUTORIZADA A REPRODUÇÃO INTEGRAL DESTA DISSERTAÇÃO APENAS PARA EFEITOS DE INVESTIGAÇÃO, MEDIANTE DECLARAÇÃO ESCRITA DO INTERESSADO, QUE A TAL SE COMPROMETE.

2. É AUTORIZADA A REPRODUÇÃO PARCIAL DESTA DISSERTAÇÃO (indicar, caso tal seja necessário, nº máximo de páginas, ilustrações, gráficos, etc.), APENAS PARA EFEITOS DE INVESTIGAÇÃO, MEDIANTE DECLARAÇÃO ESCRITA DO INTERESSADO, QUE A TAL SE COMPROMETE.

3. De acordo com a legislação em vigor, não é permitida a reprodução de qualquer parte desta dissertação

Guimarães, \_\_\_/\_\_\_/\_\_\_\_\_

Assinatura: \_\_\_\_\_

---

*“Talvez não tenha conseguido fazer o melhor, mas lutei para que o melhor fosse feito.  
Não sou o que deveria ser, mas Graças a Deus, não sou o que era antes.”*

**Marthin Luther King**



## Resumo

A implementação de sistemas automatizados na indústria implica o treino de pessoal especializado para trabalhar/desenvolver sistemas constituídos por autómatos programáveis (vulgarmente denominados como PLC's, do inglês *programmable logic controller*), sensores e atuadores de vários tipos e funções, desde o simples comando de uma válvula até complexos controladores de processos.

A principal vantagem desta plataforma de simulação é a possibilidade de fornecer aos alunos novas estratégias e metodologias de aprendizagem, tendo como base as práticas laboratoriais direcionadas ao que irão encontrar no mercado de trabalho. Através da utilização do seu computador pessoal, os alunos serão capazes de aprender ao seu próprio ritmo, autonomamente, tendo a capacidade de descobrir e incidir na resolução das suas próprias dificuldades.

Esta dissertação apresenta todo o processo de construção de plataformas de simulação virtuais de sistemas automatizados, réplicas de sistemas reais, para que o controlo do sistema possa ser simulado virtualmente utilizando a simulação *Model-In-the-Loop*. Devido às vantagens destes processos, esta plataforma foi desenvolvida no contexto do ensino da área de automação aos estudantes das várias áreas da Engenharia.

Dado isto, ao longo desta tese são enunciados todos os passos para o desenvolvimento da plataforma, assim como os formalismos e ferramentas utilizados.

### Palavras-Chave:

Ensino da Automação, Simulação de Sistemas Automatizados, Bancadas Didáticas, Model-In-the-Loop, Software-In-the-Loop, Hardware-In-the-Loop, Modelação da Parte Operativa, Autómatos Finitos Temporizados, Linguagem Ladder, CX-One.



## **Abstract**

The implementation of automated systems in the industry involves training specialized personnel to work / develop systems consisting of programmable logic controllers (commonly referred to as PLC's), sensors and actuators of various types and functions, from the simple command of a valve to complex processes controllers.

The main advantage of this simulation platform is the ability to provide students new learning strategies and methodologies, based on laboratory practices directed at what they will find in their labor market. Through the use of their personal computer, students will be able to learn at their own pace, independently, having the ability to discover and focus on solving their own problems.

This thesis presents the entire process of building platforms for virtual simulation of automated systems, replica of real systems, so that the control system can be simulated using the simulation virtually Model-In-the-Loop. Due to the advantages of these processes, this platform has been developed in the context of teaching automation to the students of the most extensive areas of Engineering.

Given that, along this thesis are listed all the steps for the development of the platform, as well as formalisms and tools used.

### **Keywords:**

Automation Teaching, Automated Systems Simulation, Model-In-the-Loop, Software-In-the-Loop, Hardware-In-the-Loop, Plant Modeling, Finite Timed Automata, Ladder Language, CX-One.



## **Agradecimentos**

Embora uma dissertação seja, pela sua finalidade acadêmica, um trabalho individual, há contributos de natureza diversa que não podem e nem devem deixar de ser realçados. Por essa razão, desejo expressar os meus sinceros agradecimentos:

Ao Professor Doutor José Machado, meu orientador, pela amizade, pela sua competência científica e partilha de conhecimento, pela disponibilidade e generosidade reveladas ao longo do desenvolvimento do trabalho, assim como pelas críticas, correções e sugestões relevantes feitas durante a orientação.

À Professora Doutora Filomena Soares, pela competência científica demonstrada, traduzida em valiosos conselhos e sugestões, bem como pela disponibilidade e amizade demonstradas.

Aos meus queridos pais, Eduardo e Orinda, e irmão, Diogo, pela forma como me acompanharam e ajudaram ao longo desta etapa da minha formação. Obrigado pelo amor, alegria e atenção ao longo de toda a minha vida, que fizeram de mim o que sou hoje.

Finalmente, um agradecimento muito especial à minha amada Daniela, pelo seu carinho e amizade, paciência e incondicional apoio que sempre demonstrou. Muito obrigado pelas conversas e palavras de coragem nos momentos menos bons, bem como pela força transmitida.

A todos, o meu MUITO OBRIGADO.



# Índice

<b>1. INTRODUÇÃO.....</b>	<b>1</b>
1.1. Contextualização e motivação .....	1
1.2. Objetivos.....	2
1.2.1. Objetivo Principal.....	2
1.2.2. Objetivos específicos.....	3
1.3. Organização da dissertação .....	3
<b>2. FUNDAMENTOS TEÓRICOS PARA A MODELAÇÃO DA PARTE OPERATIVA .....</b>	<b>7</b>
2.1. Terminologia e conceitos.....	7
2.2. Formalismos de modelação .....	9
2.2.1. Modelica.....	9
2.2.2. Redes de Petri elementares.....	10
2.2.3. Autómatos finitos temporizados.....	12
2.3. Metodologias para a modelação de componentes físicos.....	13
2.3.1. Modelação em modelos individuais instanciáveis.....	15
2.3.1.1. Exemplo de aplicação.....	15
2.3.1.2. Vantagens e desvantagens.....	20
2.3.2. Modelação em modelos de grupos instanciáveis.....	20
2.3.2.1. Exemplo de aplicação.....	21
2.3.2.2. Vantagens e desvantagens.....	23
2.3.3. Modelos observadores.....	24
2.3.3.1. Exemplo de aplicação.....	25
2.4. Informação a reter do capítulo.....	27
<b>3. SIMULAÇÃO.....</b>	<b>31</b>
3.1. Simulação de sistemas automatizados.....	31
3.1.1. Software-in-the-loop (SiL).....	32
3.1.2. Hardware-in-the-loop (HiL).....	34
3.1.3. Model-in-the-loop (MiL).....	35
3.1.4. Laboratory Testing (LT).....	36
3.2. Simuladores .....	37
3.2.1. Dymola.....	37

3.2.2.	Simulink .....	40
3.2.3.	CX-One .....	42
3.2.3.1.	<i>Cx-Programmer</i> .....	43
3.2.3.2.	<i>CX-Simulator</i> .....	44
3.2.3.3.	<i>CX-Designer</i> .....	45
3.3.	Informação a reter do capítulo .....	46
<b>4.</b>	<b>METODOLOGIA DE CONSTRUÇÃO DAS BANCADAS DE SIMULAÇÃO .....</b>	<b>49</b>
4.1.	Abordagem desenvolvida.....	49
4.2.	Dificuldades encontradas .....	55
4.3.	Informação a reter do capítulo .....	57
<b>5.</b>	<b>CASOS DE ESTUDO .....</b>	<b>61</b>
5.1.	Caso de estudo nº1 – Barreira automática.....	61
5.1.1.	Modelação do sistema automatizado.....	62
5.1.1.1.	<i>Tabelas de entradas e saídas do controlador</i> .....	63
5.1.1.2.	<i>Modelo da barreira</i> .....	63
5.1.1.3.	<i>Modelo das viaturas</i> .....	64
5.1.2.	Simulação.....	65
5.1.2.1.	<i>Alterações efetuadas nos modelos</i> .....	65
5.1.2.2.	<i>Ambiente de simulação</i> .....	66
5.2.	Caso de estudo nº2 – Misturadora de solução química.....	69
5.2.1.	Modelação do sistema automatizado.....	70
5.2.1.1.	<i>Tabelas de entradas e saídas do controlador</i> .....	71
5.2.1.2.	<i>Modelo do produto A</i> .....	73
5.2.1.3.	<i>Modelo do produto B</i> .....	74
5.2.1.4.	<i>Modelo do Solvente S</i> .....	76
5.2.1.5.	<i>Modelo da solução química</i> .....	77
5.2.2.	Simulação.....	78
5.2.2.1.	<i>Alterações efetuadas nos modelos</i> .....	78
5.2.2.2.	<i>Ambiente de simulação</i> .....	80
5.3.	Caso de estudo nº3 – Estação de transporte de objetos.....	83
5.3.1.	Modelação do sistema automatizado.....	84
5.3.1.1.	<i>Tabelas de entradas e saídas do controlador</i> .....	85
5.3.1.2.	<i>Modelos dos carros C1 e C2</i> .....	86

5.3.1.3.	<i>Modelos da garra</i> .....	89
5.3.1.4.	<i>Modelos dos objetos</i> .....	92
5.3.2.	Simulação .....	94
5.3.2.1.	<i>Alterações efetuadas nos modelos</i> .....	94
5.3.2.2.	<i>Ambiente de simulação</i> .....	96
5.4.	Aplicação das simulações ao ensino da automação .....	98
5.5	Informação a reter do capítulo .....	99
<b>6.</b>	<b>CONCLUSÃO E TRABALHOS FUTUROS</b> .....	<b>103</b>
	<b>REFERÊNCIAS</b> .....	<b>105</b>



## Índice de Figuras

Figura 1 – Representação esquemática da interação entre as partes de um sistema de automação.....	2
Figura 2 – Conceito de Sistema automatizado.....	8
Figura 3 – Circuito elétrico simples e o seu modelo construído em Modelica [11].....	10
Figura 4 – Exemplo de uma rede de Petri [12].....	11
Figura 5 – Rede de Petri marcada [12].....	12
Figura 6 – Elementos fundamentais dos autómatos finitos temporizados [15]. ....	13
Figura 7 – Diferentes granularidades para um modelo de um cilindro pneumático com a sua válvula solenoide e dois sensores de fim de curso [16]. ....	14
Figura 8 – Sistema a modelar composto por uma electroválvula monoestável, um cilindro de simples efeito e dois sensores de fim de curso.....	16
Figura 9 – Modelo da electroválvula.....	17
Figura 10 – Modelo do cilindro pneumático. ....	18
Figura 11 – À esquerda modelo do sensor A. À direita modelo do sensor B.....	19
Figura 12 - Sistema a modelar composto por uma electroválvula monoestável, um cilindro de simples efeito e dois sensores de fim de curso.....	21
Figura 13 – Modelo do grupo “electroválvula, cilindro e sensores A e B”.....	22
Figura 14 – Sistema a modelar utilizado nos exemplos anteriores, desta vez com uma peça incluída. ....	25
Figura 15 – a) Modelo do grupo electroválvula, cilindro, sensores A e B. b) Modelo observador da peça. ....	26
Figura 16 – Resumo das diferentes formas de simulação e suas principais diferenças.....	32
Figura 17 – Configuração SiL - Exemplo de vários aparelhos ligados simultaneamente ao computador onde corre o simulador [20].....	33
Figura 18 – Exemplo de simulação HiL [23].....	34
Figura 19 – Ambiente de modelização do Dymola [33]. ....	38
Figura 20 – Ambiente de simulação do Dymola [33] .....	39
Figura 21 – Arquitetura do Dymola [35]. ....	40
Figura 22 – Ambiente de trabalho do Simulink e sua biblioteca de blocos funcionais.....	41
Figura 23 – Modelo construído no Simulink de uma turbina de um aerogerador [36]. ....	42
Figura 24 - Janela principal do CX-Programmer. ....	44
Figura 25 - Janela principal do CX-Designer e suas características [48].....	45

Figura 26 – Exemplo de um enunciado e representação de um exercício. ....	50
Figura 27 – Modelos construídos para o exemplo apresentado: a) modelo do grupo “acionador, berbequim e sensores”, b) modelo individual da broca. ....	52
Figura 28 – Ambiente do CX-Programmer com as equações escritas. ....	53
Figura 29 – Ambiente de simulação criado no CX-Designer para o exemplo proposto. ....	54
Figura 30 – Exemplo de um teste realizado. ....	55
Figura 31 – Gráfico da proporção da origem dos erros surgidos na simulação ao longo do trabalho realizado. ....	57
Figura 32 – Barreira automática e sensores constituintes do sistema a modelar. ....	61
Figura 33 – Modelo do grupo “motor-barreira-sensores sbs e sbd”. ....	63
Figura 34 – Modelo observador das viaturas. ....	65
Figura 35 – Modelo das viaturas utilizado na simulação. ....	66
Figura 36 – Representação parcial das equações ladder dos modelos construídos. ....	67
Figura 37 – Ambiente de simulação criado no CX-Designer. ....	67
Figura 38 – Ambiente do CX-Designer ao longo da simulação: Viatura ultrapassou a barreira. ....	68
Figura 39 - Ambiente do CX-Designer ao longo da simulação: Viatura não ultrapassou a barreira e inverteu o sentido de marcha. ....	68
Figura 40 – Instalação de preparação da solução química. ....	69
Figura 41 – Modelo do produto A. ....	73
Figura 42 – Modelo do produto B. ....	75
Figura 43 – Modelo do solvente S. ....	76
Figura 44 – Modelo da solução química. ....	77
Figura 45 – Modelos reconstruídos através das alterações para a simulação: a) Produto A, b) Produto B. ....	79
Figura 46 – Modelo da solução corrigido. ....	80
Figura 47 - Representação parcial das equações ladder dos modelos construídos. ....	80
Figura 48 – Ambiente de simulação construído no CX-Designer. ....	81
Figura 49 – Ambiente do CX-Designer ao longo da simulação. ....	82
Figura 50 – Sistema automatizado a modelar. ....	84
Figura 51 - Modelo construído para o carro C1. ....	87
Figura 52 – Modelo construído para o carro C2. ....	88
Figura 53 – Modelo do braço da garra. ....	90

Figura 54 – Modelo do guindaste da garra. ....	91
Figura 55 – Modelo da pinça. ....	92
Figura 56 – Modelos observadores da peça: a) no carro C1, b) no carro C2, c) na grua, d) no tapete rolante.....	93
Figura 57 – Modelos do guindaste provenientes das alterações efetuadas para a simulação: a) modelo da carga, b) modelo da descarga. ....	95
Figura 58 – Modelo observador da peça na garra corrigido. ....	95
Figura 59 – Representação parcial das equações ladder dos modelos construídos. ....	96
Figura 60 – Ambiente de simulação criado no CX-Designer. ....	97
Figura 61 - Ambiente do CX-Designer ao longo da simulação. ....	98



## Índice de Tabelas

Tabela 1 - Áreas específicas onde cada um dos softwares presentes no CX-One pode ser utilizado [41].	43
Tabela 2 – Entradas do controlador correspondentes às saídas da parte física.	51
Tabela 3 – Saídas do controlador correspondentes às entradas da parte física.	51
Tabela 4 – Entradas do controlador/saídas da parte operativa.	63
Tabela 5 – Saídas do controlador/entradas da parte operativa	63
Tabela 6 – Descrição de cada um dos estados do modelo do grupo “motor-barreira-sensores sbs e sbd”	64
Tabela 7 – Descrição de cada um dos estados do modelo observador das viaturas.	65
Tabela 8 - Entradas do controlador/saídas da parte operativa.	72
Tabela 9 – Saídas do controlador/entradas da parte operativa.	72
Tabela 10 – Descrição de cada um dos estados do modelo do produto A.	73
Tabela 11 – Descrição de cada um dos estados do modelo do produto B.	75
Tabela 12 – Descrição de cada um dos estados do modelo do solvente S.	76
Tabela 13 – Descrição de cada um dos estados do modelo da solução.	77
Tabela 14 – Entradas do controlador/saídas da parte operativa.	86
Tabela 15 – Saídas do controlador/entradas da parte operativa.	86
Tabela 16 – Descrição de cada um dos estados do modelo do carro C1.	87
Tabela 17 – Descrição de cada um dos estados do modelo do carro C2.	88
Tabela 18 – Descrição de cada um dos estados do modelo do braço da garra.	90
Tabela 19 - Descrição de cada um dos estados do modelo do guindaste da garra.	91
Tabela 20 – Descrição de cada um dos estados do modelo da pinça da garra.	92
Tabela 21 – Descrição de cada um dos estados dos modelos observadores da peça.	93



## **Lista de Abreviaturas e Siglas**

HiL – Hardware-in-the-loop

HMI – Human-machine Interface

IEC – International Electrotechnical Commission

LT – Laboratory Testing

MiL – Model-in-the-loop

PC – Personal Computer

PLC – Programmable Logic Controller

RdP – Redes de Petri

SiL – Software-in-the-loop

USB – Universal Serial Bus



# Capítulo 1

# INTRODUÇÃO



# 1. INTRODUÇÃO

## 1.1. Contextualização e motivação

A implementação do tratado de Bolonha originou uma série de alterações nos processos de ensino e aprendizagem utilizados nas universidades. A sucessiva redução de horas de contato entre professores e alunos levaram a uma reformulação das metodologias de ensino, fazendo com que o aluno tenha um papel cada vez mais importante na sua aprendizagem [1]. Além de ter de se concentrar nas aulas lecionadas pelos professores, o aluno deve ter uma atitude ativa fora delas. Ele deve trabalhar fora do horário escolar, com o objetivo de aprender o máximo sobre os conteúdos lecionados e desenvolver as suas competências.

Na área da engenharia a forma mais eficaz de um aluno aprender, é praticando [2]. Para que possa corresponder às exigências de um mercado de trabalho cada vez mais competitivo, é importante que o aluno tenha na sua formação um prévio contato com situações semelhantes das que irá encontrar na indústria. Como é sabido, as universidades possuem laboratórios onde é possível simular as situações previamente referidas. No entanto, é praticamente impossível todos os alunos terem total acesso a eles devido ao seu reduzido espaço, falta de pessoal qualificado e ao elevado preço de alguns dos componentes que se vão desgastando com o seu manuseamento.

Para tornar possível aos alunos pôr os seus conhecimentos em prática sem a necessidade de recorrer à utilização dos laboratórios, podem ser utilizados ambientes virtuais de simulação. Além de permitirem trabalhar e resolver exercícios práticos em qualquer altura e em qualquer lugar, os ambientes virtuais de simulação têm a capacidade de transmitir um *feedback* imediato, permitindo ao aluno trabalhar e aprender ao seu próprio ritmo [3]. Estes devem ser de fácil utilização, tendo apenas o requisito de o aluno ter instalado no seu computador o *software* adequado para as operações de simulação.

Para que os alunos da área da automação possam usufruir das vantagens referidas no parágrafo anterior, são necessários ambientes de simulação específicos capazes de corresponder às necessidades da sua aprendizagem. Por este motivo torna-se evidente que é de extrema importância que seja feita uma investigação com o propósito de criar e melhorar este tipo de ferramentas.

## 1.2. Objetivos

Atendendo aos argumentos referidos no subcapítulo anterior, foram traçados objetivos bem definidos a cumprir ao longo deste trabalho. Nos pontos seguintes são apresentados em detalhe, juntamente com as suas justificações.

### 1.2.1. Objetivo Principal

O objetivo principal deste trabalho é a elaboração de bancadas virtuais de simulação destinadas aos alunos de automação. Estas bancadas baseiam-se em exercícios práticos animados onde os alunos, seguindo enunciados distribuídos nas aulas pelos seus professores, deverão ser capazes de desenvolver e implementar de forma bem-sucedida programas para o controlador. Analisando a simulação, estes obtêm um *feedback* imediato, permitindo-lhes uma aprendizagem autónoma à medida que detetam e corrigem os seus próprios programas. Visto ser lecionado nas aulas e ser gratuito para os alunos, é utilizado o *software* CX-One da marca OMRON para a elaboração das bancadas e a simulação dos programas construídos.

Para desenvolver as bancadas de simulação será necessário modelar a parte operativa e definir a interação entre esta e a parte do controlador, como representado na Figura 1.

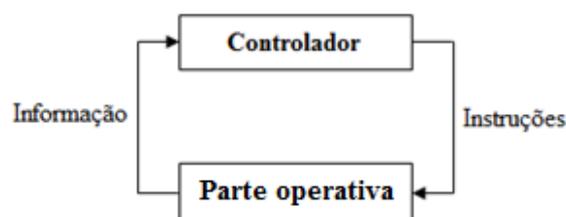


Figura 1 – Representação esquemática da interação entre as partes de um sistema de automação.

A interação entre os dois modelos (controlador e parte operativa) é modelada através de variáveis Booleanas e não-Booleanas, que vão servir de base de construção dos respetivos modelos. Para modelar a parte operativa são utilizados formalismos específicos para o efeito.

### **1.2.2. Objetivos específicos**

Com o objetivo de deixar ferramentas úteis para a continuação deste trabalho ou desenvolvimento de outros a partir deste, são estabelecidos dois objetivos específicos a atingir.

O primeiro baseia-se em encontrar e desenvolver uma metodologia eficaz para a construção de modelos de componentes físicos a partir dos enunciados dos exercícios, e conseqüentemente a elaboração das suas bancadas de simulação. É importante que seja estabelecida uma metodologia padrão, que possa servir como um guia a ser facilmente seguido por utilizadores com menor experiência neste tipo de modelação, na resolução de outros tipos de exercícios além dos apresentados ao longo desta dissertação.

Por fim, o segundo objetivo específico proposto consiste em construir uma biblioteca de módulos instanciáveis composta por modelos virtuais de componentes físicos construídos ao longo do desenvolvimento das bancadas de simulação. Esta biblioteca de módulos instanciáveis pode vir a ser uma ferramenta muito útil para o futuro, pois irá permitir o acesso a todos os modelos de componentes físicos construídos, podendo estes ser reutilizados para modelar comportamentos de diferentes sistemas, mas com módulos similares.

### **1.3. Organização da dissertação**

O texto desta dissertação encontra-se num formato que melhor permite atingir os objetivos delineados. Dessa forma, em termos gerais, está bipartido numa primeira parte que aborda os conceitos teóricos estudados e necessários de assimilar para a compreensão dos capítulos seguintes (capítulos 2 e 3), seguindo-se de uma segunda parte onde são apresentados resultados do trabalho desenvolvido (capítulos 4 e 5).

Globalmente, a dissertação encontra-se organizada em 6 capítulos distintos.

No capítulo 1 efetua-se o enquadramento da dissertação na temática do ensino da automação, interesse do tema e respetivos objetivos definidos que motivaram a sua execução.

No capítulo 2 apresentam-se os fundamentos teóricos essenciais para a construção dos modelos de um sistema automatizado. Numa primeira fase, faz-se referência a alguns dos formalismos existentes para a modelação de componentes físicos, seguindo-se de uma explicação sobre os tipos de abordagens distintas que podem ser adotadas para a construção dos modelos.

No capítulo 3 são expostos os conceitos fundamentais sobre simulação. São apresentados e explicados os diferentes tipos de abordagens de simulação atualmente existentes, fazendo referência às suas principais diferenças. Como complemento, são também enunciados alguns dos simuladores mais utilizados na indústria e no ensino universitário.

No capítulo 4 é definida a metodologia desenvolvida ao longo do trabalho para a elaboração das bancadas de simulação. Este processo é aqui dividido em múltiplos passos simples, para o tornar sistemático e mais perceptível ao leitor.

No capítulo 5 são resolvidos três casos de estudo com grau de dificuldade crescente. Em cada caso de estudo, numa primeira fase é apresentada a sua resolução pormenorizada, seguindo-se da sua simulação no *software* selecionado para o efeito.

O capítulo 6 apresenta as conclusões gerais da dissertação e propostas de desenvolvimento de trabalhos futuros neste domínio.

Por fim, apresentam-se as referências bibliográficas.

## **Capítulo 2**

# **FUNDAMENTOS TEÓRICOS PARA A MODELAÇÃO DA PARTE OPERATIVA**



## 2. FUNDAMENTOS TEÓRICOS PARA A MODELAÇÃO DA PARTE OPERATIVA

Este capítulo tem como finalidade apresentar os fundamentos teóricos necessários para a compreensão do trabalho desenvolvido ao longo desta dissertação.

É definido o conceito de sistema automatizado, bem como seus constituintes e interações necessárias para o funcionamento do sistema. Relativamente à modelação da sua parte operativa, são abordados alguns exemplos do grande número de formalismos de modelação de componentes físicos existentes, e apresentado o formalismo de modelação utilizado para a construção das bancadas de simulação – Autómatos Finitos Temporizados.

Para concluir este capítulo, são apresentadas as metodologias de modelação de modelos instanciáveis desenvolvidas ao longo da dissertação. Através da sua aplicação a um exemplo ilustrativo, são explicadas as suas diferenças e enunciadas as suas principais vantagens e desvantagens.

### 2.1. Terminologia e conceitos

Um sistema automatizado é um sistema mecatrónico composto por três blocos funcionais distintos, indispensáveis para o seu bom funcionamento [4]. Os blocos funcionais possuem as seguintes funções:

- Interface Homem-máquina (HMI) – do inglês *Human-machine Interface*, é o painel de instrumentos da máquina, onde se encontram os botões, luzes de aviso, etc. É através deste painel que o utilizador pode atuar no sistema, através de botões, pedais ou alavancas;
- Controlador – podendo ser um PLC, lógica cablada, PC ou um microcontrolador, é o responsável pelo controlo do sistema. Mediante a avaliação dos sinais provenientes pelo HMI e pela parte operativa (entradas), vai emitir ordens de comando (saídas).
- Parte operativa – Conjunto de componentes físicos existentes numa máquina. Estes encontram-se divididos entre atuadores e transdutores. Os atuadores são os

elementos que recebem diretamente as ordens emitidas pelo controlador e são responsáveis por fazer com que a máquina se mova e realize a operação desejada. Os transdutores (sensores) são os componentes físicos responsáveis por transmitir o mais variado tipo de informação ao controlador (temperatura, pressão, posicionamento, etc.).

Os blocos funcionais referidos no parágrafo anterior interagem entre si através de sinais lógicos, como se encontra representado na Figura 2.

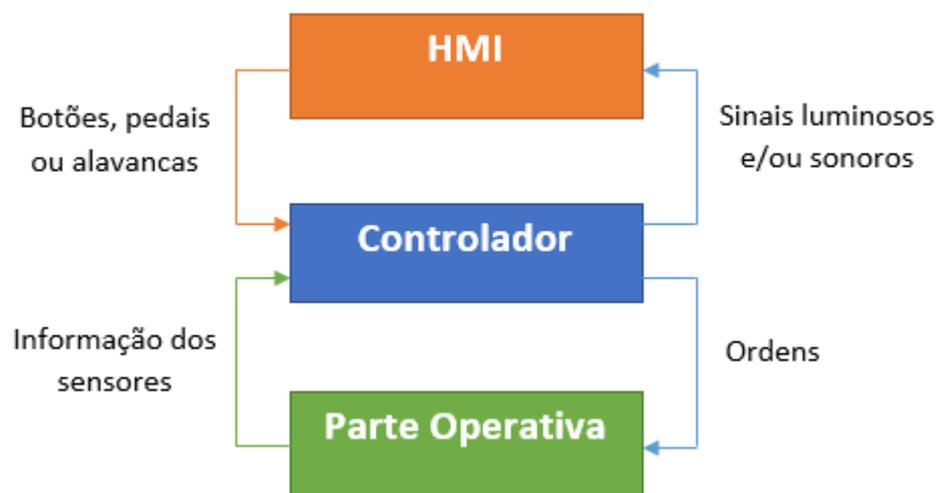


Figura 2 – Conceito de Sistema automatizado.

Uma experiência pode ser vista como um processo de extração de dados de um sistema, através da excitação das suas entradas e observação das suas saídas. Por comportamento do sistema entende-se a forma como o sistema reage às entradas gerando as saídas [5]. 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.

A descrição do comportamento dos sistemas físicos pode ser sistematizada numa forma mais abstrata através de um modelo. Um modelo pode ser definido como sendo um conjunto de equações ou declarações utilizadas para descrever o comportamento de um sistema [6]. Os modelos matemáticos de um sistema físico podem, por si só, fornecer informação sobre o sistema, mas são maioritariamente utilizados para prever o seu comportamento futuro. O projeto 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 projeto de plataformas de teste para o sistema real [5].

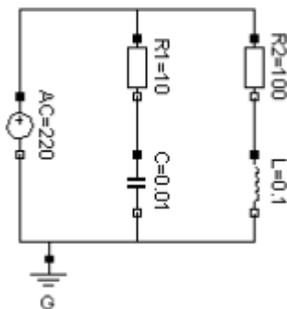
## 2.2. Formalismos de modelação

### 2.2.1. Modelica

Modelica [7] é uma linguagem de modelação, muitas vezes designada por “linguagem de descrição de *hardware*”. Este formalismo permite a especificação de sistemas físicos complexos em modelos matemáticos, para a simulação computacional de sistemas dinâmicos em que o seu comportamento evolui em função do tempo.

Através da sua utilização em *softwares* de simulação como o Dymola [8], a linguagem Modelica é utilizada numa ampla gama de tarefas de simulação em sistemas mecânicos, termodinâmicos, elétricos, automóveis, hidráulicos, entre outros. Isto levou ao desenvolvimento de uma grande variedade de bibliotecas contendo um grande número de modelos de diversos componentes dos vários sistemas referidos, disponíveis gratuitamente através de páginas dedicadas como a *Modelica Association* [9] ou *OpenModelica* [10].

Na Figura 3 encontra-se representado um pequeno circuito elétrico e o seu modelo construído utilizando a linguagem Modelica. O circuito é constituído por uma fonte de tensão, duas resistências, uma bobina, um condensador e um ponto neutro (terra). Os modelos destes componentes encontram-se normalmente disponíveis em bibliotecas, podendo ser seleccionados e utilizados diretamente nos *softwares* específicos para o efeito [11].



```

model circuit
  Resistor R1 (R=10);
  VsourceAC AC;
  Capacitor C (C=0.01);
  Ground G;
  Resistor R2 (R=100);
  Inductor L (L=0.1);
equation
  connect(AC.p, R1.p); // Capacitor circuit
  connect(R1.n, C.p);
  connect(C.n, AC.n);
  connect(R1.p, R2.p); // Inductor circuit
  connect(R2.n, L.p);
  connect(L.n, C.n);
  connect(AC.n, G.p);
end circuit;

```

Figura 3 – Circuito elétrico simples e o seu modelo construído em Modelica [11].

No formalismo Modelica, os modelos são construídos utilizando a linguagem de programação C ou suas variantes, tal como o C++. Inicialmente são definidos os componentes constituintes do sistema. Por exemplo, na Figura 3, a declaração “Resistor R1 (R=10);” declara a R1 um componente do tipo Resistência e define o seu valor padrão como sendo  $10\Omega$ . As conexões são declaradas de seguida, responsáveis por estabelecer as ligações entre os componentes do modelo. Uma conexão deve conter todas as grandezas necessárias para estabelecer uma interação. No caso de componentes elétricos, devem ser definidas dentro das conexões as grandezas a tensão (V) e a corrente (I). Para cada uma das grandezas devem ser indicados os valores inicial, mínimo e máximo admissíveis para completar o modelo do sistema.

### 2.2.2. Redes de Petri elementares

As redes de petri (RdP) são um formalismo matemático que pode ter uma representação gráfica; fornece um ambiente uniforme para a modelação, análise formal e simulação de sistemas a eventos discretos, permitindo uma visualização simultânea da sua estrutura e do seu comportamento [12].

A representação gráfica de uma RdP básica é formada por três componentes básicos [13]:

- Um círculo (ou elipse), denominado *lugar*. Este componente denota algo passivo, tal como um estado local de um componente, uma condição, um recurso partilhado, etc.;
- Uma barra (ou retângulo), denominada *transição*. Este componente denota algo ativo, tal como um evento, uma ação, uma transição, uma transmissão de mensagem, etc.;
- Arcos dirigidos responsáveis por unir os lugares e as transições e determinar as causas e os efeitos. Um arco entre um lugar e uma ação indica que essa condição é necessária para a executar (pré-condição). Por sua vez, Um arco entre uma ação e um lugar indica que a sua execução irá tornar essa condição verdadeira (pós-condição).

Na Figura 4 encontra-se representado um exemplo de uma RdP, contendo todos os seus elementos básicos.

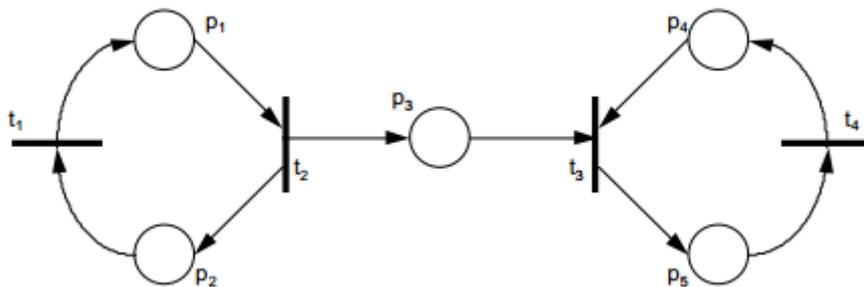


Figura 4 – Exemplo de uma rede de Petri [12].

A estrutura de uma rede de Petri pode ser definida como um tuplo  $R = (L, T, AE, AS)$ , onde:

- $L = \{ p_1, p_2, \dots, p_m \}$  é um conjunto de lugares;
- $T = \{ t_1, t_2, \dots, t_m \}$  é um conjunto de transições;
- $L \cap T = \emptyset$  os conjuntos L e T são disjuntos;
- $AE = L \times T$  é um conjunto de arcos de entrada nas transições;
- $AS = T \times L$  é um conjunto de arcos de saída das transições.

Dada uma RdP constituir um modelo abstrato, podem ser obtidas várias interpretações distintas a lugares e transições, conforme o tipo de aplicação em causa. No entanto, dadas as suas características intrínsecas, os lugares podem, de uma forma muito

genérica e muito próxima da sua visualização gráfica, ser vistos como depósitos de recursos e as transições como ações que manipulam esses recursos. Os recursos são representados graficamente por pequenos círculos pretos dentro dos lugares. A cada um desses círculos dá-se o nome de *marca*. A função das transições consiste em destruir e/ou criar estas marcas. Como as transições estão obrigatoriamente entre lugares, é através da sua ação (denominada disparo) que um lugar altera a sua marcação. Os arcos indicam, para cada transição, os lugares sobre os quais estas atuam. Na figura 5 encontra-se um exemplo de uma RdP marcada.

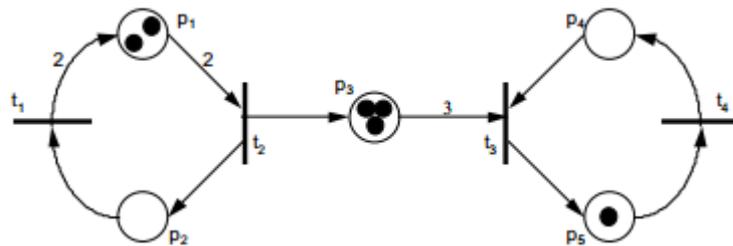


Figura 5 – Rede de Petri marcada [12].

Sendo as marcações nos lugares as representantes dos estado da rede, as transições surgem como os agentes que fazem a rede evoluir de estado para estado.

### 2.2.3. Autómatos finitos temporizados

Os autómatos finitos temporizados são máquinas de estados finitos que conseguem modelar o tempo, através da aplicação de relógios (*clocks*). Estes podem ser utilizados na modelação e análise do comportamento ao longo de tempo de sistemas computadorizados, como por exemplo sistemas em tempo real ou redes [14].

Os elementos estruturais constituintes dos autómatos finitos temporizados são: estado, guarda, invariante, transição, mensagem e atribuições de variáveis, tal como se encontra ilustrado na Figura 6.

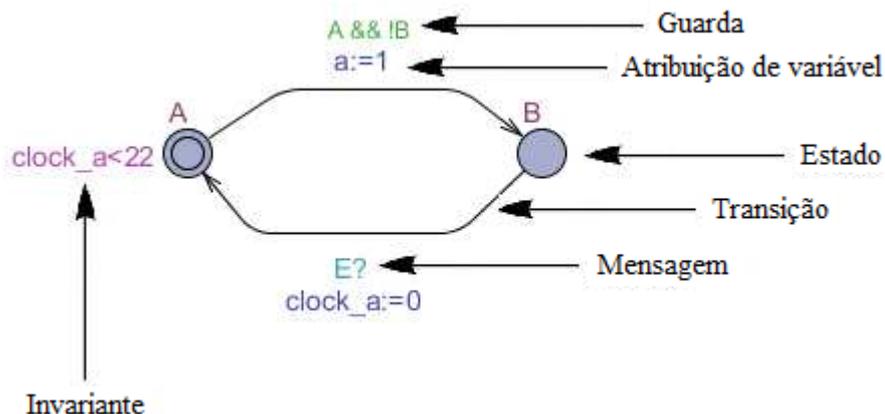


Figura 6 – Elementos fundamentais dos autômatos finitos temporizados [15].

O estado indica onde o mecanismo atual se encontra no seu processo de evolução. Cada transição encontra-se rotulada por uma equação lógica designada guarda, que indica a condição em que a transição poderá ser disparada. Podem ser atribuídos valores às variáveis do sistema quando a transição é disparada, através das condições de atribuição de variável. As invariantes (condições que têm de ser sempre respeitadas) são utilizadas juntamente com os relógios. Estas indicam o tempo máximo que aquele estado pode estar ativo, quando a transição para o estado seguinte é verdadeira. As mensagens são utilizadas para ligar e sincronizar diferentes autômatos em evolução, quando se pretende ter evoluções simultâneas de dois ou mais módulos do sistema [15].

### 2.3. Metodologias para a modelação de componentes físicos

Constata-se que os modelos físicos dos constituintes dos sistemas automatizados podem ser construídos de várias formas distintas, tendo em conta o nível de abstração e a granularidade escolhidos para os modelos [16], como se encontra representado na Figura 7.

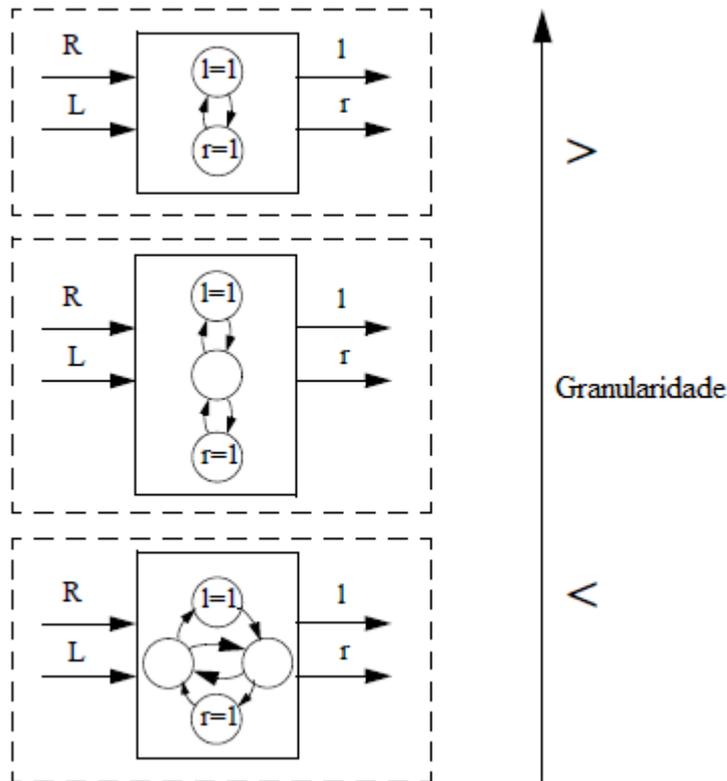


Figura 7 - Diferentes granularidades para um modelo de um cilindro pneumático com a sua válvula solenoide e dois sensores de fim de curso [16].

A escolha do grau de abstração e da granularidade dos modelos tem implicação direta nos resultados obtidos na utilização das técnicas de análise de controladores industriais, mais concretamente na utilização da simulação e da verificação formal. Alguns aspetos, que podem ser referidos, são:

- Recursos informáticos e tempo disponíveis – Menor granularidade implica recursos informáticos mais robustos e maiores intervalos de tempo necessários para a obtenção de resultados;
- Propriedades que se desejam provar – Se o utilizador quiser provar propriedades simples dos modelos (por exemplo, provar se um sensor se encontra “ligado” ou “desligado” quando acontecem determinadas eventos), modelos com menor número de estados devem ser utilizados. No entanto, quando se quiser provar propriedades mais complexas, tais como o avanço ou o recuo de um cilindro pneumático situado na sua posição intermédia, ao fim de o controlador dar uma determinada ordem, modelos mais complexos e com maior número de estados devem ser utilizados.

Nos subcapítulos seguintes são apresentadas três das principais metodologias existentes e utilizadas na modelação dos componentes físicos ao longo da dissertação. Recorrendo ao mesmo exemplo serão explicadas as suas diferenças, tais como as suas respetivas vantagens e desvantagens.

### **2.3.1. Modelação em modelos individuais instanciáveis**

Um modelo individual instanciável é definido como um modelo que representa apenas o comportamento do componente físico a que está associado. Utilizando uma metodologia assente neste tipo de modelos implica que seja necessário modelar cada um dos constituintes do sistema, para que cada um deles possua o seu modelo individual. As guardas das transições da maioria dos modelos consideram estados de outros modelos, fazendo com que exista uma dependência entre os modelos. Assim um modelo só irá passar de um estado para o outro quando os modelos de que depende permitirem.

Na modelação em modelos individuais instanciáveis, os modelos construídos podem ser separados em dois grupos distintos:

- Modelos dos atuadores – Modelos que recebem diretamente as ordens emitidas pelo controlador e que vão influenciar o comportamento dos modelos físicos a que estão ligados. Exemplos: motores, electroválvulas;
- Modelos influenciados pelos atuadores – Modelos em que o seu comportamento depende dos atuadores a que estão ligados. Exemplos: cilindro, barreira.

De seguida é apresentado um exercício ilustrativo onde pode ser entendido com mais pormenor o que foi escrito no parágrafo anterior.

#### **2.3.1.1. Exemplo de aplicação**

Pretende-se modelar os componentes físicos constituintes do sistema automatizado apresentado na Figura 8. O sistema contém uma electroválvula direcional monoestável 3/2 com retorno por mola, que ao receber a ordem V1 faz avançar um cilindro pneumático de simples efeito. Existem também dois sensores de fim de curso, A e B, que detetam quando o cilindro se encontra recuado ou avançado, respetivamente.

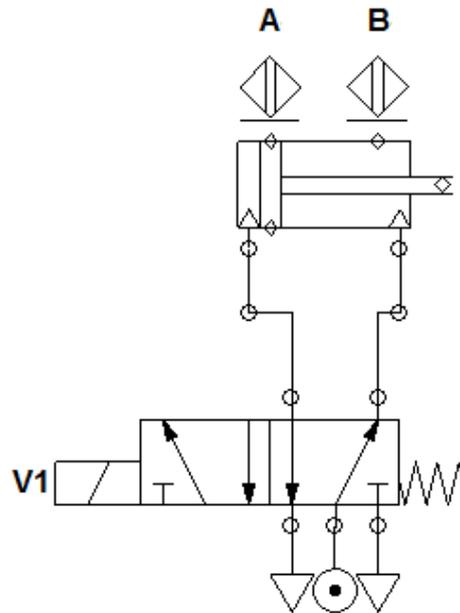


Figura 8 – Sistema a modelar composto por uma electroválvula monoestável, um cilindro de simples efeito e dois sensores de fim de curso.

Como já foi referido anteriormente, aplicando o método de modelação em modelos individuais instanciáveis, é necessário construir o modelo individual de cada um dos componentes físicos. Assim, são necessários construir quatro modelos:

- Electroválvula

Visto que o comportamento da electroválvula depende apenas da existência (ou não) da ordem V1 emitida pelo controlador, este componente físico foi o primeiro a ser modelado pois, diretamente ou indiretamente este modelo irá influenciar todos os outros modelos do sistema representado na figura anterior. Na Figura 9 encontra-se representado o modelo da electroválvula elaborado.

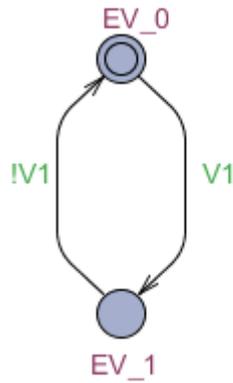


Figura 9 – Modelo da electroválvula.

Para simular o comportamento da electroválvula, foi considerado que um modelo de 2 estados é o suficiente. No modelo, estados EV\_0 e EV\_1 correspondem à posição inicial e à posição quando a ordem V1 se encontra ativa, respetivamente.

No modelo, a transição dos estados “EV\_0” para “EV\_1” ocorre instantaneamente quando a ordem V1 é ativada. No entanto, na realidade as coisas não acontecem dessa forma. A electroválvula real necessita de um curto intervalo de tempo (na ordem dos décimos de segundo ou até mesmo dos segundos) para trocar de posição, que vai ser maior ou menor dependendo de vários fatores externos.

- Cilindro pneumático

Na Figura 10 apresentada de seguida, encontra-se representado o modelo elaborado para o cilindro pneumático.

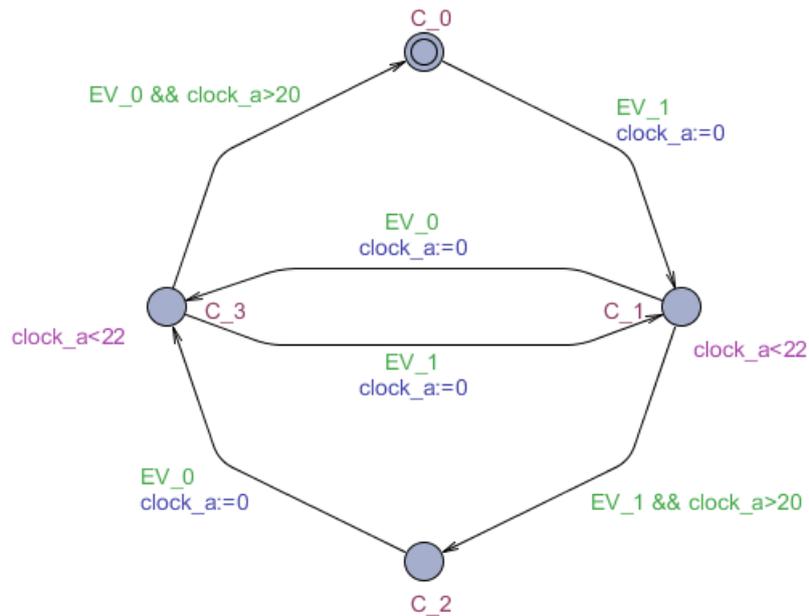


Figura 10 – Modelo do cilindro pneumático.

Para simular o comportamento do cilindro pneumático, foi utilizado um modelo de quatro estados. Ao contrário do modelo da electroválvula apresentado anteriormente, no modelo do cilindro pneumático foi considerado o tempo necessário para o seu avanço e o seu recuo, tendo sido acrescentado relógios nas guardas das transições para “C\_0” e “C\_2”.

Os estados “C\_0” e “C\_2” correspondem às posições estáticas do cilindro, ou seja recuado e avançado, respetivamente. Por outro lado, os estados “C\_1” e “C\_3” correspondem às posições dinâmicas – “C\_1” indica quando o cilindro está a avançar e “C\_3” indica quando este se encontra a recuar.

Este modelo é totalmente dependente do modelo da electroválvula. Como pode ser visto na Figura 10, o cilindro só irá avançar quando o estado EV\_1 do modelo da electroválvula (ver Figura 9) estiver ativo. Por outro lado, o cilindro só irá recuar quando o estado EV\_0 estiver ativo.

- Sensores A e B

Na Figura 11 podem ser visualizados os modelos elaborados para simular o comportamento dos sensores A e B.

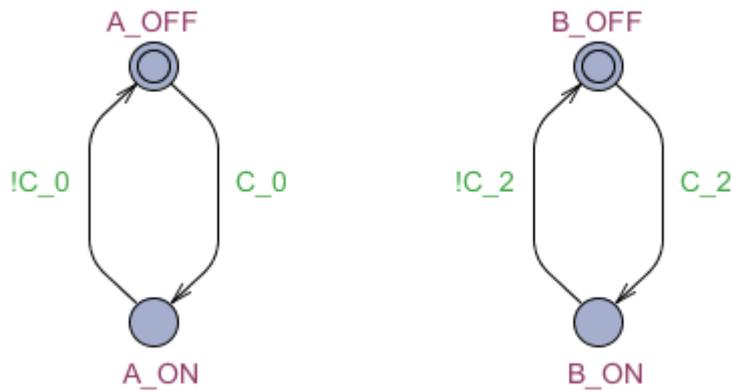


Figura 11 – À esquerda modelo do sensor A. À direita modelo do sensor B.

Como representado na figura anterior, os modelos elaborados para simular o comportamento dos sensores são semelhantes ao modelo utilizado para a electroválvula. São modelos de 2 estados em que “OFF” e “ON” correspondem a sensor desligado e ligado, respetivamente.

Tal como o modelo cilindro é dependente do modelo da electroválvula (como foi visto anteriormente), os modelos dos sensores são dependentes do modelo do cilindro (ver Figura 10). Assim:

- O sensor A liga quando o cilindro se encontra totalmente recuado (estado “C\_0”), desligando-se quando o cilindro começar a avançar (estado “C\_1”);
- O sensor B liga quando o cilindro se encontra totalmente avançado (estado “C\_2”), desligando-se quando o cilindro começa a recuar (estado “C\_3”).

Como no caso da electroválvula, os sensores reais não se ligam instantaneamente como acontece nos modelos. No entanto o tempo necessário para os sensores reais ligarem e desligarem é muito reduzido (muito inferior ao tempo necessário para a electroválvula mudar de posição). Esse tempo poderia ser simulado acrescentando estados e guardas com relógios nas transições, mas visto que não interfere com o processo de simulação, foi ignorado.

### **2.3.1.2. Vantagens e desvantagens**

A modelação por modelos individuais é fácil de implementar e reúne as seguintes vantagens:

- Simplicidade dos modelos – Cada componente físico possui o seu modelo, facilitando a construção de modelos mais simples;
- Reutilização dos modelos instanciáveis – Todos os modelos apresentados são instanciáveis e podem ser reutilizados na modelação de outros componentes similares;
- Facilidade na organização dos modelos – Como os modelos dos componentes físicos são individuais, podem ser facilmente organizados em grupos (por exemplo, atuadores, transdutores, etc.).

Apesar de possuir as vantagens suprarreferidas e de ser uma ferramenta útil e eficaz para modelar os componentes físicos da maioria dos sistemas automatizados, esta metodologia possui as suas desvantagens, tais como:

- Número elevado de modelos – Para sistemas que possuem um elevado número de componentes físicos, a modelação utilizando esta metodologia pode tornar-se morosa e enfadonha, aumentando a probabilidade de ocorrerem erros de modelação;
- Difícil deteção de erros – Como os modelos são dependentes uns dos outros, quando ocorre um erro é necessário verificar todos os modelos construídos para detetar o erro, corrigi-lo e evitar a sua propagação;
- Elevado consumo de recursos informáticos e tempo na verificação formal – À medida que o número de modelos aumenta, o consumo de recursos informáticos e tempo necessários para a sua verificação formal aumentam exponencialmente, pois todas as possibilidades da atividade de estados têm de ser calculadas.

### **2.3.2. Modelação em modelos de grupos instanciáveis**

Um modelo de um grupo, instanciável, é um modelo que é utilizado para simular o comportamento de vários componentes físicos, simultaneamente. Agrupando os componentes dependentes uns dos outros num único modelo, este é construído para

minimizar ao máximo o número de modelos necessários para simular o comportamento de um sistema. Estes modelos são mais compactos que os modelos individuais, sendo por isso mais complexos de construir. As suas transições vão possuir nas guardas ordens dadas pelo controlador, podendo também conter relógios para simular o tempo. Ao contrário da metodologia apresentada anteriormente, as transições poderão também possuir atribuição de valores a variáveis para permitir a simulação do comportamento de todos os componentes físicos.

De seguida é apresentado o mesmo exercício ilustrativo utilizado anteriormente, mas modelado com a metodologia de modelação em modelos de grupos instanciáveis.

### 2.3.2.1. Exemplo de aplicação

Pretende-se modelar os componentes físicos constituintes do sistema apresentado na Figura 12. O sistema contém uma electroválvula direcional monoestável 3/2 com retorno por mola, que ao receber a ordem V1 faz avançar um cilindro pneumático de simples efeito. Existem também dois sensores de fim de curso, A e B, que detetam quando o cilindro se encontra recuado ou avançado, respetivamente.

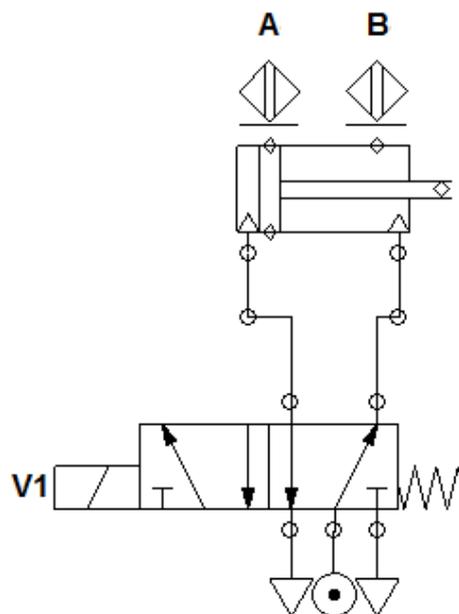


Figura 12 - Sistema a modelar composto por uma electroválvula monoestável, um cilindro de simples efeito e dois sensores de fim de curso.

No sistema representado na figura anterior, estão representados quatro componentes físicos: electroválvula, cilindro pneumático e dois sensores de fim de curso. Mais uma vez, e lembrando o que já foi escrito, é possível observar que o comportamento dos componentes é dependente do comportamento da electroválvula. Atendendo a este fato é possível juntá-los num único grupo, podendo assim se construir um único modelo de todo o sistema.

Na Figura 13 está representado o modelo elaborado seguindo a metodologia de modelação em modelos de grupos instanciáveis.

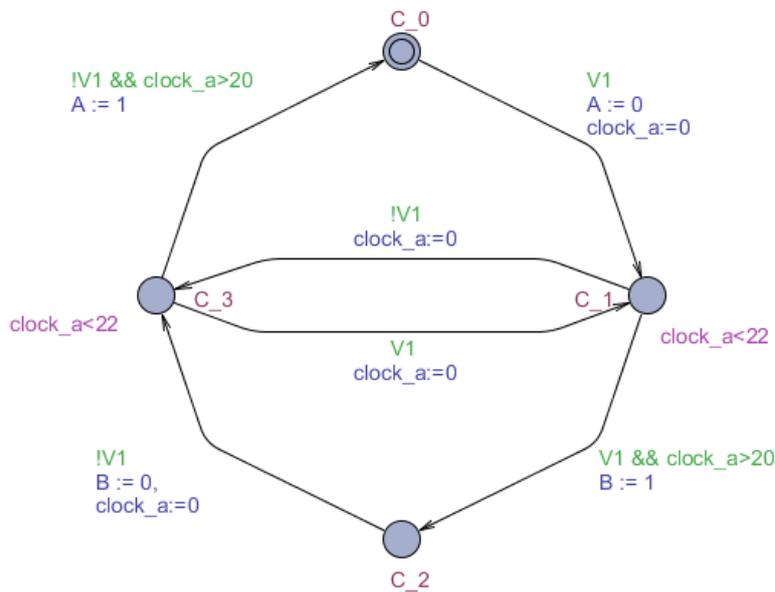


Figura 13 – Modelo do grupo “electroválvula, cilindro e sensores A e B”.

O cilindro pneumático é o único atuador (componente que realiza trabalho) do grupo. Por esse motivo o modelo foi construído para que cada estado corresponda a uma posição do cilindro:

- C\_0 – Cilindro recuado;
- C\_1 – Cilindro a avançar;
- C\_2 – Cilindro avançado;
- C\_3 – Cilindro a recuar.

Como pode ser visto nas transições do modelo, dependendo da guarda V1, este vai alterar o seu estado – Quando V1 existe, o modelo vai avançar na direção do estado “C\_2” e quando não existe, vai recuar em direção ao estado “C\_0”. Além disto, estão

também presentes as atribuições de variáveis que têm como função “ligar” e “desligar” os sensores A e B:

- O sensor A liga quando existe a transição de “C\_3” para “C\_0”, desligando-se quando existe a transição de “C\_0” para “C\_1”;
- O sensor B liga quando existe a transição de “C\_1” para “C\_2”, desligando-se quando existe a transição de “C\_2” para “C\_3”.

Visto que na realidade o cilindro não passa da posição recuada para a avançada e vice-versa instantaneamente, relógios foram adicionados às guardas para simular o tempo necessário à concretização dos seus movimentos.

### **2.3.2.2. Vantagens e desvantagens**

A modelação por modelos de grupos possui as seguintes vantagens:

- Número reduzido de modelos – Mesmo para sistemas com grande número de componentes físicos, um menor número de modelos são construídos para os simular;
- Modelos instanciáveis – Podem ser reutilizados na modelação de outros componentes (grupos) similares;
- Reduzido consumo de recursos informáticos e tempo de simulação – Visto os modelos serem em menor número e mais compactos que os modelos individuais, os programas de simulação necessitam de menor tempo e memória para executarem todos os cálculos necessários na verificação formal;
- Detecção de erros mais facilitada – Caso exista um erro de modelação, com esta metodologia é mais fácil a sua deteção, pois o modelo do grupo não funciona da forma como deveria.

Além das vantagens suprarreferidas, a modelação por modelos de grupo instanciáveis têm as suas desvantagens, tais como:

- Complexidade dos modelos – Em alguns casos os modelos podem tornar-se bastante complexos, aumentando a probabilidade de acontecerem erros na sua modelação;
- Verificação formal restringida – Previamente à construção deste tipo de modelos, as propriedades que se querem ver provadas na verificação formal devem ser bem definidas. Caso não se tenha este cuidado, corre-se o risco de existirem restrições na verificação formal dos modelos construídos impostas pelos componentes físicos que não foram diretamente modelados.

### **2.3.3. Modelos observadores**

Os modelos observadores são modelos independentes do controlador, e como o próprio nome indica, são construídos para “observar” determinados comportamentos ao longo da simulação. Este tipo de modelos são ideais para modelar peças, fluídos, etc. cujos comportamentos vão depender das posições dos componentes físicos a eles associados. Os modelos observadores podem ser divididos em duas categorias distintas:

- Modelos observadores de bom funcionamento – Modelos construídos para simular o comportamento pretendido de um objeto. São modelos mais simples de construir, sendo maioritariamente sequenciais;
- Modelos observadores de erro – Modelos construídos para simular comportamentos de “erro” de um objeto em situações reais. São modelos muito complexos e morosos de construir, pois têm de conter todos os comportamentos de “erro” possíveis de acontecer na realidade para poderem ser simulados.

Ao longo da dissertação foram utilizados apenas modelos observadores de bom funcionamento devido à sua maior facilidade de construção. Além disso, os modelos observadores de erro estão fora do âmbito desta dissertação. De seguida é apresentado um exercício ilustrativo da utilização deste tipo de modelos.

### 2.3.3.1. Exemplo de aplicação

Pretende-se modelar os componentes físicos constituintes do sistema apresentado na Figura 14. O sistema contém uma electroválvula direccional monoestável 3/2 com retorno por mola, que ao receber a ordem V1 faz avançar um cilindro pneumático de simples efeito. Existem também dois sensores de fim de curso, A e B, que detetam quando o cilindro se encontra recuado ou avançado, respetivamente.

Uma peça de cada vez é colocada por um operador, que irá ativar o cilindro para a empurrar para dentro de uma caixa. O cilindro deve recuar imediatamente e o processo repetido sempre que seja colocada uma nova peça pelo operador.

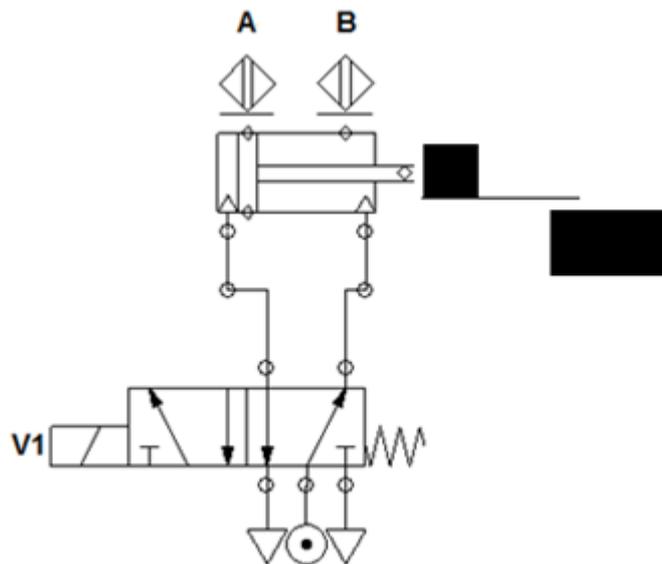


Figura 14 – Sistema a modelar utilizado nos exemplos anteriores, desta vez com uma peça incluída.

Como pode ser observado na Figura 14, além dos componentes físicos (electroválvula, cilindro pneumático, sensores A e B) encontra-se também representada uma peça que vai ser transportada até à caixa através do avanço do cilindro. Para simular o comportamento dos componentes físicos é utilizado o modelo de grupo já construído anteriormente. Este modelo será o responsável pelas transições do modelo observador construído para simular o comportamento da peça.

Na Figura 15 encontram-se representados os modelos de grupo e da peça, respetivamente.

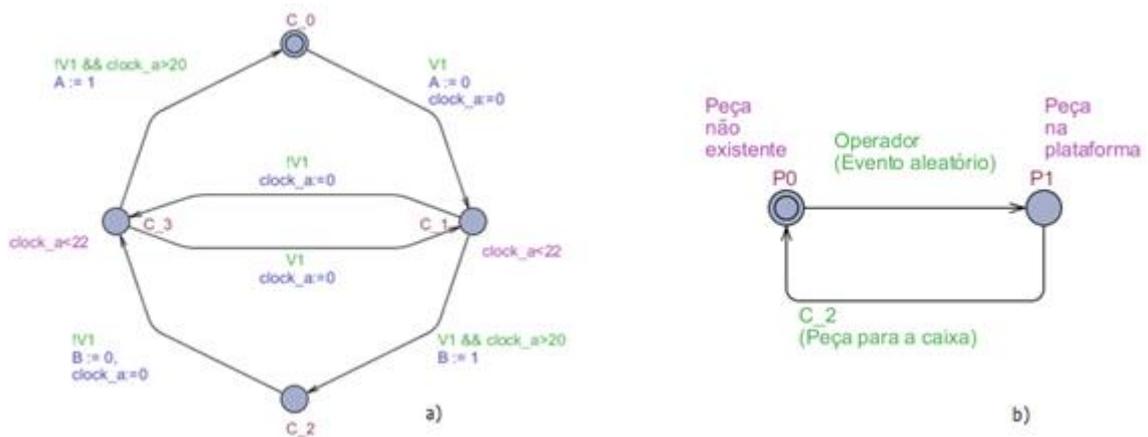


Figura 15 – a) Modelo do grupo electroválvula, cilindro, sensores A e B. b) Modelo observador da peça.

Como pode ser visto na Figura 15, o modelo observador da peça é um modelo sequencial simples constituído por dois estados. O estado inicial “P0” indica que a peça não existe no sistema, enquanto o estado “P1” indica que a peça já foi colocada pelo operador e está pronta para ser movida para a caixa.

Simular a ação realizada pelo operador por si só, é bastante complexo, pois não é possível estabelecer um padrão a seguir para o comportamento de um ser Humano. Visto que o objetivo do exercício não é esse, foi utilizado um evento aleatório gerado por *software* como solução para o comportamento deste. Quando este evento for “verdadeiro”, significa que a peça vai estar inserida na plataforma, ativando então o estado “P1”.

A peça vai manter-se na plataforma até ser empurrada para a caixa pelo cilindro. Isto só vai acontecer quando o cilindro atingir a sua posição avançada, isto é, o estado “C\_2”. Quando esta condição for verdadeira, deixa então de haver peça na plataforma, voltando o estado “P0” do modelo observador a ficar ativo.

O exercício apresentado foi um exemplo simples que explica de uma maneira muito superficial a construção e o funcionamento de um modelo observador de bom funcionamento. No entanto, para exercícios mais complexos estes modelos podem não ser tão simples de construir. Dependendo das propriedades que o utilizador quiser provar, os modelos observadores de bom funcionamento podem conter um variado número de estados, além de poder possuir outras funcionalidades importantes tais como ligar e desligar sensores (células de carga, por exemplo).

## 2.4. Informação a reter do capítulo

As interações entre um sistema e o seu ambiente são de dois tipos, definidas como entradas e saídas, dependendo da sua origem e onde vão produzir o seu efeito.

Existem vários formalismos com diferentes características que podem ser utilizados na modelação de sistemas automatizados. Cada um deles é visto como uma ferramenta diferente para chegar ao mesmo objetivo, que é a construção de um modelo matemático para a representação do comportamento do sistema.

Ao longo deste trabalho foram desenvolvidas e utilizadas três metodologias diferentes na modelação dos sistemas automatizados. Por trazer vantagens relativamente ao número de modelos necessários para representar o sistema e serem necessários menores recursos informáticos na utilização de técnicas de análise, optou-se por utilizar sempre que possível a modelação em modelos de grupos instanciáveis de componentes físicos.



# Capítulo 3

# SIMULAÇÃO



### **3. SIMULAÇÃO**

A simulação é uma técnica difundida um pouco por todo o mundo, utilizada nas mais diversas aplicações, tais como no ensino acadêmico, ambiente industrial e manutenção de equipamentos.

Ao longo deste capítulo é apresentado o conceito de simulação e as suas principais vantagens que a tornam cada vez mais uma ferramenta essencial para a área dos sistemas automatizados. São ainda referidas e explicadas as diferentes abordagens com que a simulação pode ser realizada, enunciando as suas aplicações e principais semelhanças e diferenças.

Por fim, além do simulador escolhido (CX-One da OMRON), são também apresentados outros simuladores existentes no mercado que poderiam ser utilizados para o desenvolvimento deste trabalho.

#### **3.1. Simulação de sistemas automatizados**

O processo de simulação consiste na utilização de expressões matemáticas ou especificações mais ou menos formalizadas, com o objetivo de imitar um processo ou uma operação do mundo real. Utilizando a definição descrita em [17], “a simulação é um processo de projetar um modelo computacional de um sistema real e conduzir experiências com este modelo, com o propósito de entender seu comportamento e/ou avaliar estratégias para sua operação”. Pode-se então dizer que a simulação é um processo amplo que engloba não apenas a construção do modelo, mas também todo o método experimental realizado na procura de:

- Descrever o comportamento do sistema;
- Construir teorias e hipóteses considerando as observações efetuadas;
- Usar o modelo para prever o comportamento futuro, isto é, prever os efeitos produzidos pelas alterações nas variáveis e/ou parâmetros no sistema.

No âmbito da engenharia, a simulação é utilizada para diversos objetivos distintos, tais como a simulação do comportamento de sistemas mecânicos, interações de conjuntos de componentes mecânicos e instalações de produção até simulação de sistemas mais

complexos, tais como *softwares* de controlo de sistemas automatizados, podendo ser consideradas diferentes ferramentas de simulação. Várias abordagens podem ser utilizadas, no entanto o objetivo continua a ser o mesmo: prevenir danos de maior envergadura e ter a certeza que, antes da execução no controlador real, que o sistema irá cumprir o comportamento esperado [18].

A simulação pode ser executada de quatro formas distintas, como representado na Figura 16. Elas designam-se por *Software-in-the-loop* (SiL), *Hardware-in-the-loop* (HiL), *Laboratory-testing* (LT) e *Model-in-the-loop* (MiL), respetivamente.

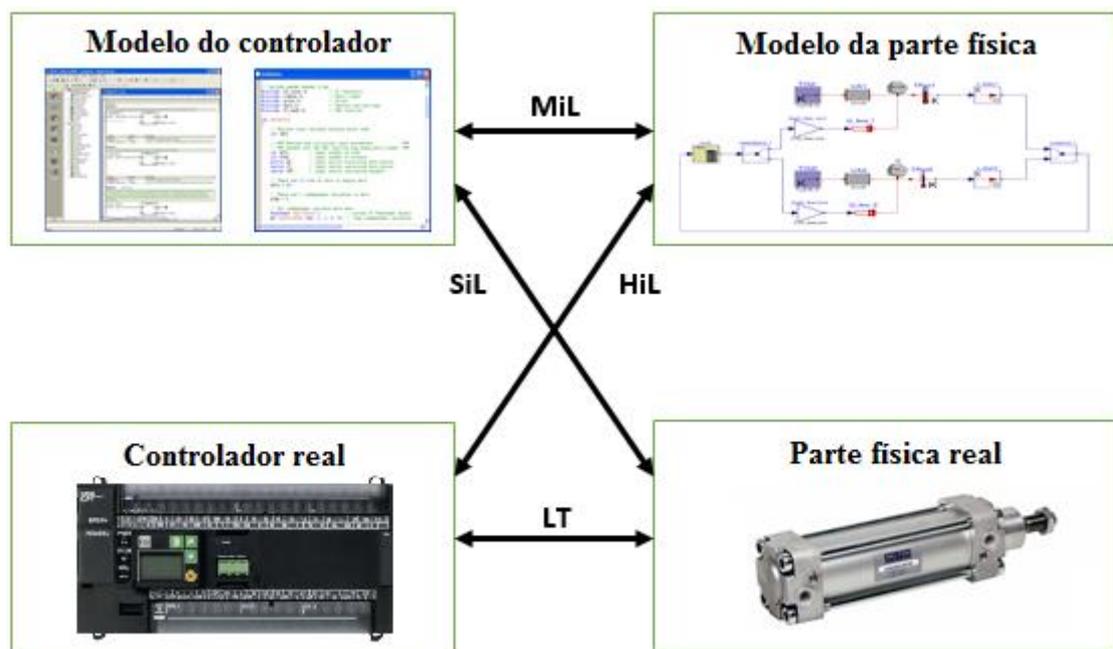


Figura 16 – Resumo das diferentes formas de simulação e suas principais diferenças.

Cada uma das diferentes formas de simulação possui as suas vantagens e desvantagens, podendo ser utilizadas individualmente ou em conjunto para permitir obter melhores resultados.

### 3.1.1. *Software-in-the-loop* (SiL)

A simulação SiL caracteriza-se por o programa do controlador ser convertido em modelos de simulação, que depois de ligados à parte física real, são simulados em um circuito fechado [19]. SiL combina a flexibilidade e o baixo custo da simulação, com a fidelidade de um emulador de *hardware*. Pode ser facilmente utilizada na conceção do

projeto e nas fases de teste dos programas, sendo por isso capaz de oferecer uma redução significativa dos custos, maximizando a reutilização de código e minimizando o esforço do seu desenvolvimento [20].

Numa configuração SiL toda a bancada de simulação pode encontrar-se num ambiente controlado, localizado num laboratório. Vários aparelhos, localizados no próprio laboratório ou fora deste, podem ser ligados simultaneamente ao computador onde o simulador está a ser executado através de uma rede *wireless* ou de outro tipo semelhante, como representado na Figura 17. Isto permite que sejam possíveis realizar um leque enorme de testes que não seriam possíveis com as formas mais tradicionais de simulação, tais como testes de campo em tempo real ou testes em sistemas de maior complexidade.

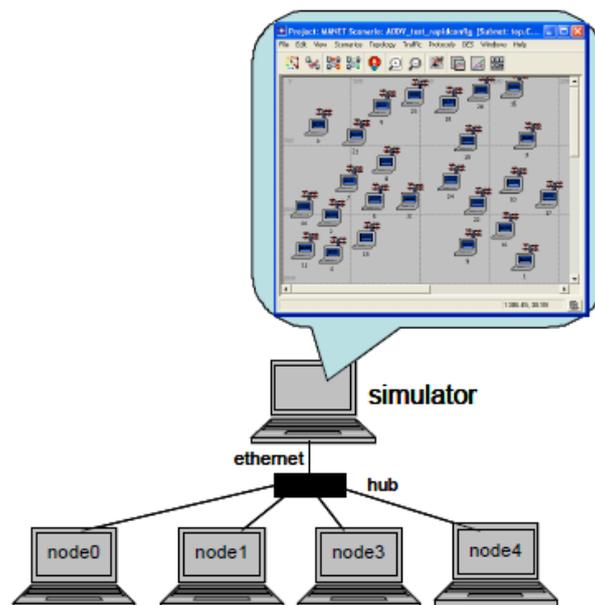


Figura 17 – Configuração SiL - Exemplo de vários aparelhos ligados simultaneamente ao computador onde corre o simulador [20].

Comparativamente com outras abordagens, a simulação SiL é uma ferramenta que permite realizar facilmente testes de simulação em ambientes controlados com grande flexibilidade e repetibilidade, a grandes velocidades de processamento (permitidas pela evolução dos processadores contidos nos computadores), sendo assim possível obter resultados bastante fiáveis num intervalo de tempo reduzido. Além disto, permite também a comparação de resultados obtidos através da simulação de diferentes estratégias, para que seja possível escolher a melhor solução para resolver um determinado problema [21].

No entanto, possui também algumas desvantagens. Muitos simuladores, principalmente os mais baratos, não possuem a capacidade de simular com rigor o tempo real. Isto pode ser um problema para simulações em que o tempo seja importante, podendo levar uma discrepância nos resultados obtidos na simulação e o que acontece na realidade. Outro fator importante é que à medida que aumenta a complexidade dos sistemas e/ou dos testes realizados sobre estes, são necessários computadores e *softwares* mais potentes. Estes normalmente têm custos bastante elevados, sendo por isso inacessíveis a empresas que não tenham capacidade de realizar este tipo de investimento.

### 3.1.2. *Hardware-in-the-loop* (HiL)

A simulação HiL é uma técnica de simulação utilizada no desenvolvimento, teste e monitorização de sistemas embebidos [22]. Esta requer a construção de uma simulação em tempo real, que modela algumas partes do sistema a testar e as interações mais significativas com o ambiente operacional. A simulação monitoriza os sinais de saída do sistema em teste (ordens para os atuadores e informação operacional para os *displays*) e injeta sinteticamente sinais de entrada gerados (sinais provenientes dos sensores e comandos provenientes dos atuadores) nos *timings* apropriados. Assim sendo, as saídas do sistema servem como entradas para a simulação, e esta gera saídas que irão ser as entradas do sistema. A Figura 18 apresenta a representação em alto nível de um exemplo de simulação HiL [23].

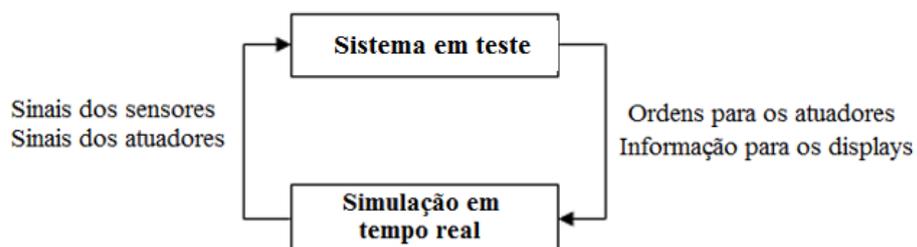


Figura 18 – Exemplo de simulação HiL [23].

Os sistemas HiL permitem a simulação de várias situações e ambientes que seriam difíceis e/ou de elevados custos financeiros de realizar, utilizando protótipos reais [24]. Eles são uma tecnologia importante no desenvolvimento de sistemas de controlo

eletrónico dentro dos prazos estabelecidos, pois facilitam os testes automatizados, permitindo que sejam realizados em maior número e que necessitam de menores períodos de tempo para a obtenção de resultados. Assim, HIL é uma ferramenta capaz de aumentar os requisitos de segurança, reduzir os custos e aumentar a reutilização de sistemas e componentes já testados e comprovados [25].

Comparada com a abordagem SiL, a abordagem HiL possui algumas vantagens devido ao fato da simulação ser realizada com o programa real a correr no controlador real. Isto pode ser muito importante, especialmente, quando são consideradas diferentes escalas de tempo de um ponto de vista real, sobretudo no que diz respeito da evolução do controlador real e a sua escala de tempo, correndo nas condições de trabalho reais. A interação entre o controlador real e o modelo da parte física, que ocorre no simulador do computador, é ainda complexa e difícil de ser executada, principalmente quando a simulação e o comportamento em tempo real são críticos para a garantia dos resultados obtidos para o sistema testado [26].

### **3.1.3. *Model-in-the-loop (MiL)***

A abordagem de simulação MiL é caracterizada pela simulação dos modelos do programa do controlador e da parte operativa em ambiente virtual. Recorrendo ao mesmo ambiente de simulação ou a ambientes de simulação diferentes, os dois tipos de modelos são testados em conjunto, permitindo obter resultados da interação entre ambas as partes do sistema.

A simulação MiL é amplamente utilizada na fase inicial do desenvolvimento de projetos de controladores, utilizados para o controlo de sistemas reais. Ao longo da simulação, o modelo do controlador comunica com o modelo da parte física, enviando sinais com ordens para os modelos dos atuadores, recebendo sinais com informação proveniente da parte física modelada (sinais de sensores, por exemplo) [27].

Este tipo de simulação é a financeiramente mais económica, já que não são necessários componentes reais para o seu funcionamento. Possuindo o simulador adequado, é possível modelar ambas as partes do sistema (controlador e parte física) e

obter resultados da simulação que ajudam a compreender o comportamento do sistema no seu funcionamento real [28].

### **3.1.4. *Laboratory Testing (LT)***

O Laboratory Testing (teste em laboratório) é uma técnica de simulação que consiste em realizar ensaios num ambiente controlado, sobre o controlador e a parte operativa reais. Esta abordagem de simulação é frequentemente utilizada nas fases finais do desenvolvimento dos projetos de sistemas, sendo que nesta fase, já são conhecidos resultados obtidos através de outros tipos de abordagens de simulação.

Nesta abordagem, os testes são realizados nas condições esperadas de funcionamento onde o sistema irá exercer a sua função. Normalmente, são realizados testes específicos de comportamento em situações de normal funcionamento e testes de segurança, como por exemplo no caso de existir algum tipo de falha, mau funcionamento de algum componente, etc. Desta forma, são obtidas informações muito próximas sobre o comportamento dos sistemas a operar em condições reais, sendo ainda possível realizar melhoramentos e corrigir os problemas encontrados no protótipo testado.

Uma das desvantagens do método experimental é que, para um grande número de sistemas, muitos dos fatores não são acessíveis ou controláveis. Estes sistemas estão sob a influência de entradas inacessíveis, designadas por perturbações. Do mesmo modo, ocorre muitas vezes o caso de muitas saídas realmente úteis não serem acessíveis para medições, designados também por estados internos do sistema [29].

Há também um certo número de problemas práticos associados com a realização de um teste de laboratório, como por exemplo:

- A experiência pode ser muito cara: investigando a durabilidade do casco de um navio através da construção de navios e deixá-los colidir é um método muito caro de obter informações;
- A experiência pode ser muito perigosa: realizar testes de laboratório simulando centrais nucleares não é aconselhável, podendo originar graves acidentes;
- O sistema necessário para a experiência pode ainda não existir. Isto é frequente acontecer em sistemas que ainda não foram concebidos ou fabricados.

## 3.2. Simuladores

A utilização de simuladores é bastante difundida, tanto no ensino universitário como na indústria. Através da utilização destes, é possível realizar inúmeros testes aos sistemas, com custos inerentes muito mais reduzidos do que recorrendo aos testes práticos [30].

Os simuladores encontram-se divididos em dois tipos distintos: simuladores de tempo contínuo e simuladores de eventos discretos. Os simuladores de tempo contínuo devem ser utilizados para simulações em que os valores das variáveis alterem continuamente, ao longo da simulação. Por outro lado, os simuladores de eventos discretos são adequados para a simulação de problemas cujos estados são definidos pela ocorrência de eventos [31]. Atualmente, devido ao grande desenvolvimento nesta área, a maioria dos simuladores são híbridos, isto é, possuem a capacidade de simular as duas configurações.

De seguida são apresentados três exemplos de simuladores bastante utilizados na simulação de sistema automatizados. É importante referir que os simuladores não se resumem apenas aos exemplos aqui apresentados. Existe uma grande variedade disponível no mercado, desenvolvidos por diferentes fabricantes e adequados a finalidades específicas. Os simuladores apresentados tiveram como único critério de escolha a sua maior exposição comparativamente com outros, ao longo do ensino universitário.

### 3.2.1. Dymola

Dymola – *Dynamic Model Laboratory* – é uma poderosa ferramenta desenvolvida pela Dynasim (mais tarde adquirida pela Dassault Systèmes), utilizada na modelação e simulação de sistemas integrados e complexos utilizados nas indústrias automóvel, aeroespacial, robótica, e de processamento, entre outras [32]. Esta é uma ferramenta baseada em modelação de componentes, que representam as peças e dispositivos reais pertencentes a um sistema, tais como engrenagens, válvulas, eixos, tubos, etc. As interações entre estes componentes são convenientemente descritas por conexões gráficas, que modelizam a sua ligação física.

A interface gráfica do utilizador do Dymola encontra-se dividida em duas janelas: ambiente de modelação (onde se realizam as construções dos modelos) e ambiente de simulação (onde se realizam as simulações e se obtém e comparam resultados) [33], representados nas Figuras 19 e 20, respetivamente.

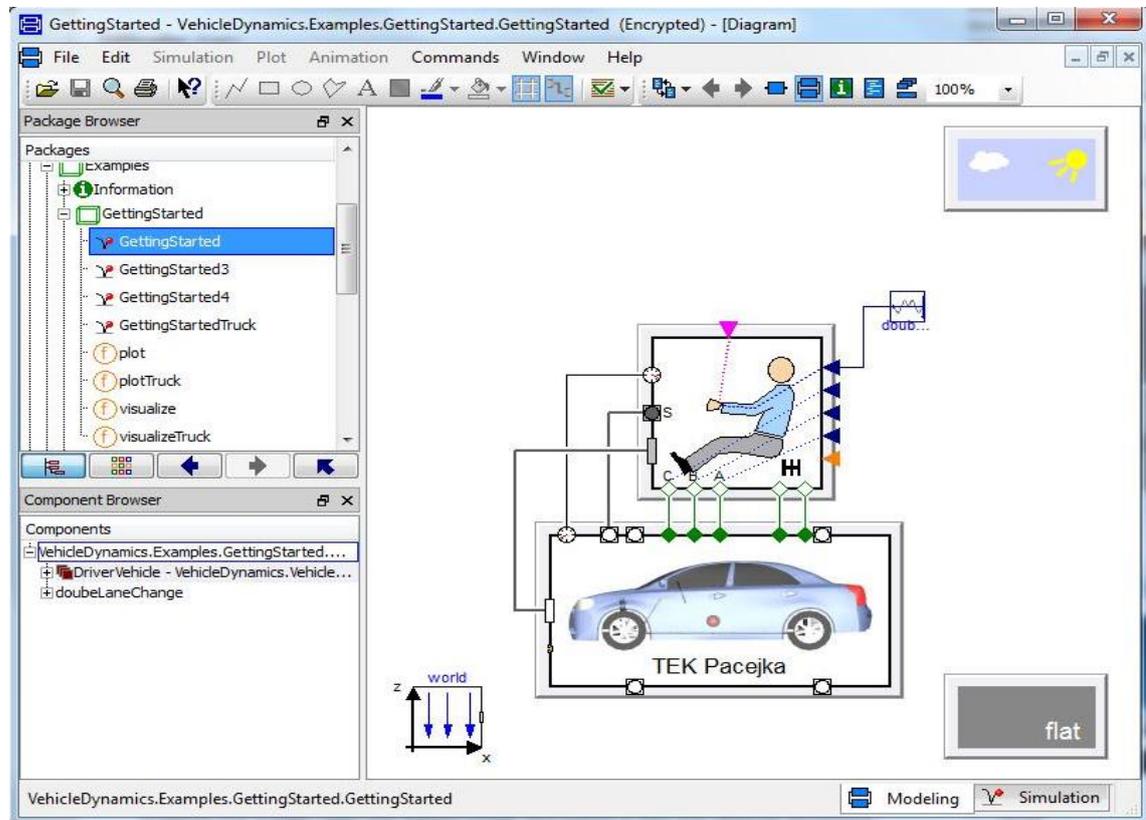


Figura 19 – Ambiente de modelação do Dymola [33].

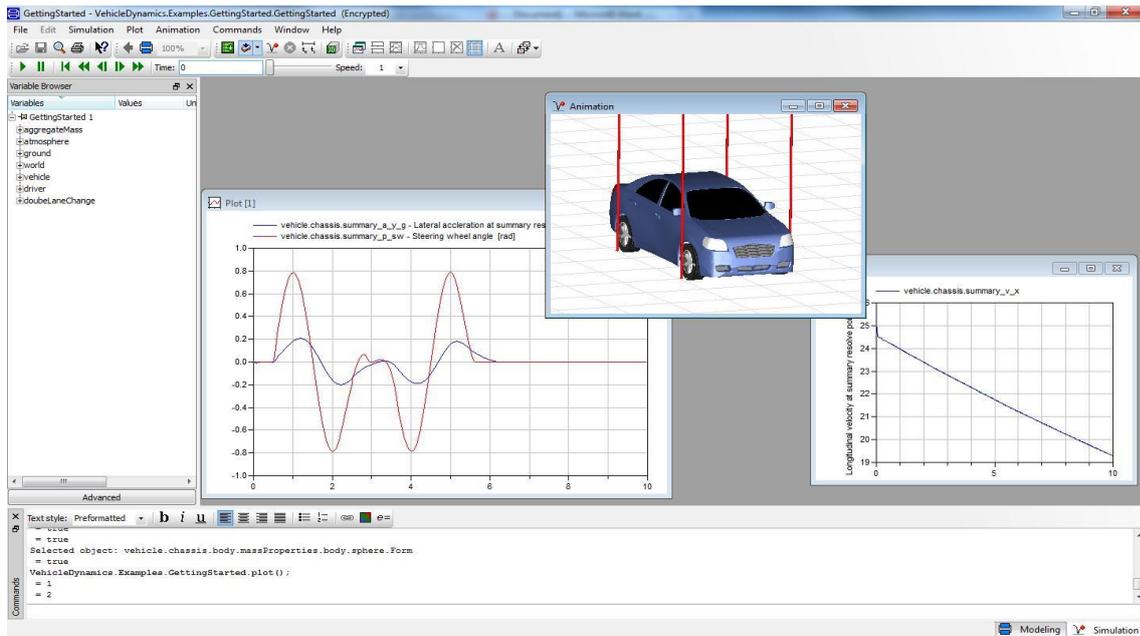


Figura 20 – Ambiente de simulação do Dymola [33]

Os componentes utilizados no Dymola são definidos utilizando o formalismo de modelação Modelica. Existe uma grande variedade de bibliotecas desenvolvidas acessíveis gratuitamente na Internet, mas caso o utilizador não encontre o componente específico necessário para a simulação em nenhuma destas bibliotecas, o Dymola permite a edição de modelos em linguagem Modelica, permitindo a sua construção. Devido a estas facilidades, esta ferramenta é amplamente utilizada na simulação de sistemas mecânicos, elétricos, termodinâmicos e hidráulicos [34].

Na Figura 21 encontra-se representado um esquema da arquitetura do Dymola. Através deste, é possível resumir e apresentar todas as potencialidades desta poderosa ferramenta de simulação.

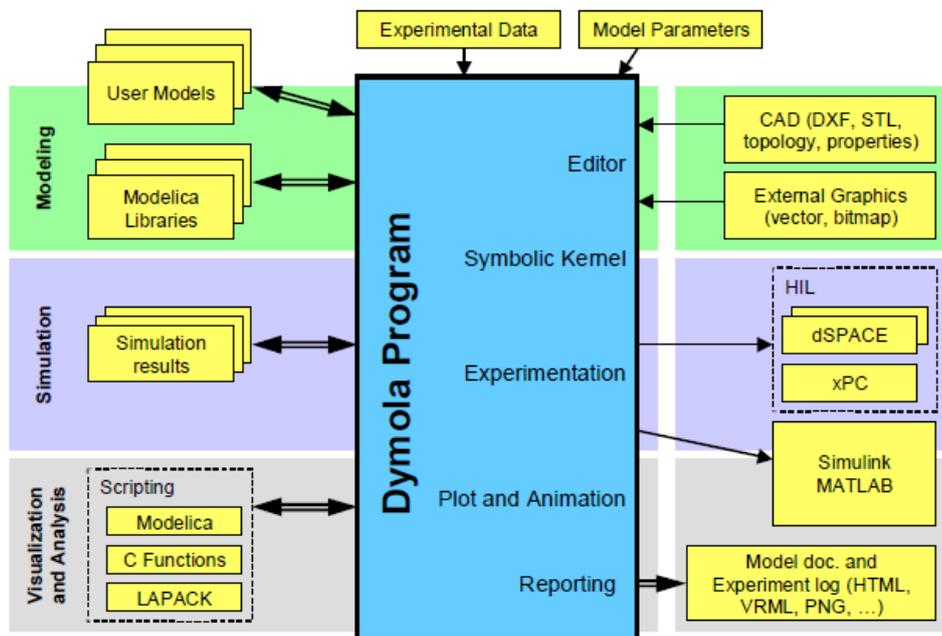


Figura 21 – Arquitetura do Dymola [35].

O Dymola permite a importação de gráficos e ficheiros de outros programas. Ele possui um tradutor simbólico integrado para equações Modelica gerando código C para a simulação. O código C pode ser exportado para o Simulink ou para plataformas de simulação *hardware-in-the-loop* [35].

### 3.2.2. Simulink

O Simulink [36], desenvolvido pela MathWorks[37], é um ambiente de simulação baseado em diagramas de blocos utilizado na modelação, simulação e análise de sistemas dinâmicos de várias áreas científicas. Ele incorpora um editor gráfico, bibliotecas de blocos personalizáveis e alguns exemplos de resoluções, que servem de base para os utilizadores com menor experiência, modelizar e simular sistemas dinâmicos.

O Simulink encontra-se integrado no *software* MATLAB, permitindo-lhe incorporar os algoritmos do MATLAB nos modelos, e exportar os resultados de simulação para a sua análise no ambiente do MATLAB. Na Figura 22 encontra-se representado o ambiente de trabalho do Simulink, juntamente com a sua biblioteca de blocos funcionais.

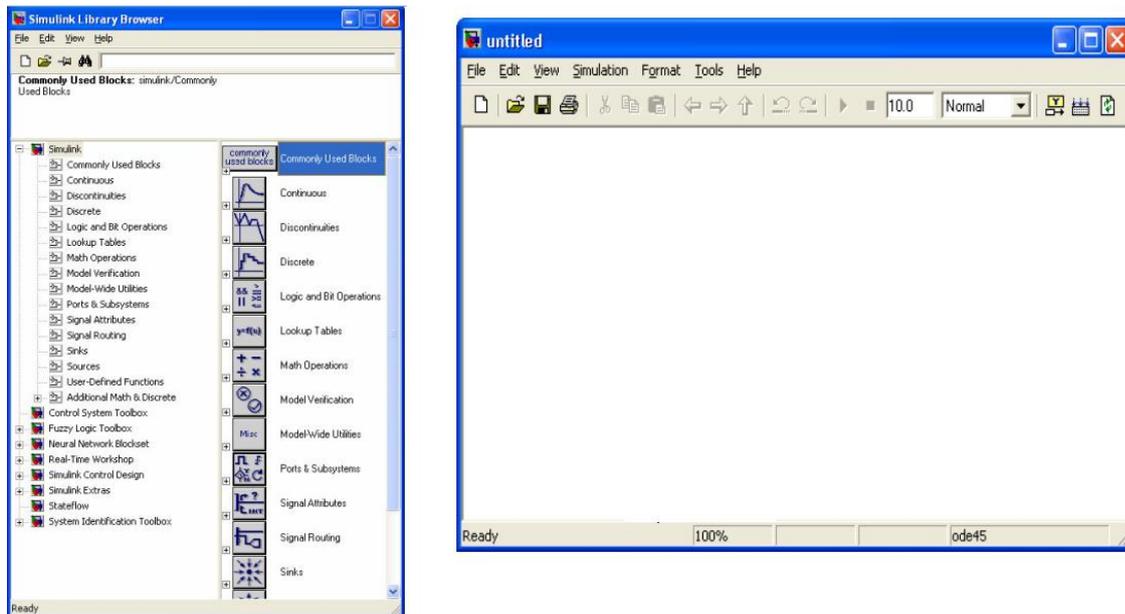


Figura 22 – Ambiente de trabalho do Simulink e sua biblioteca de blocos funcionais.

Este simulador é utilizado numa grande variedade de aplicações em diferentes áreas (tais como controlo de sistemas, processamento digital de sinal, sistemas embebidos, sistemas mecatrónicos, etc.) devido às suas potencialidades:

- Fácil e rápida construção de modelos devido à natureza “*drag and drop*” dos blocos funcionais;
- Obtenção e análise de resultados ao longo que corre o processo de simulação do sistema;
- Facilidade no manuseamento de grande quantidade de informação no projeto;
- Capacidade de conexão com *hardware* para testes em tempo real e implantação de sistemas integrados (*Hardware-in-the-loop*).

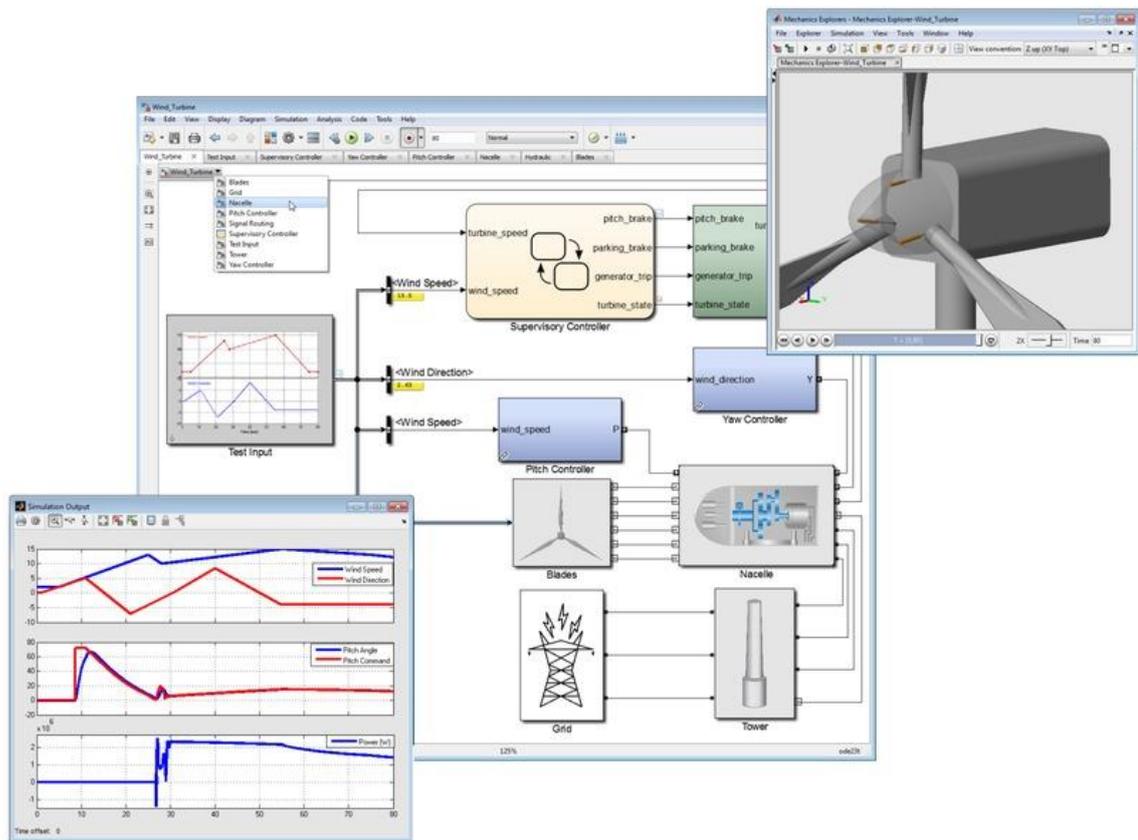


Figura 23 – Modelo construído no Simulink de uma turbina de um aerogerador [36].

### 3.2.3. CX-One

O CX-One [38] é um ambiente de programação, simulação e configuração que permite ao utilizador construir, simular ou configurar redes, PLCs, HMIs, sistemas de controlo de movimentos, motores, controladores de temperatura e sensores. Este *software* desenvolvido pela marca japonesa OMRON [39], é o resultado do agrupamento de um conjunto de ferramentas com reduzida complexidade de configuração no mesmo produto, para permitir a programação ou configuração de sistemas de automação sem haver necessidade de longos períodos de formação [40].

A Tabela 1 apresenta uma lista das áreas específicas de cada um dos *softwares* incluídos no CX-One.

<b>Áreas específicas</b>	<b>Softwares</b>
Simulação e Programação	CX-Programmer CX-Simulator CX-Designer
Rede	CX-Integrator CX-ConfiguratorFDT CX-Protocol Network Configurator
Movimento, Controladores e Motores	CX-Motion, Pro, NCF, MCH CX-Drive CX-Position Network Configurator
Regulação e Comutação	CX-Thermo CX-Process
Sensores	CX-Sensor

Tabela 1 - Áreas específicas onde cada um dos softwares presentes no CX-One pode ser utilizado [41].

Para o desenvolvimento das bancadas de simulação foram utilizados os *softwares* de simulação e programação contidos no CX-One, nomeadamente o CX-Programmer, o CX-Simulator e o CX-Designer. De seguida são apresentados pequenos resumos do funcionamento de cada um deles.

### 3.2.3.1. Cx-Programmer

O CX-Programmer é uma ferramenta de programação capaz de criar, testar e fazer a manutenção de programas para todos os tipos de PLCs fabricados pela marca OMRON [42]. Este programa possui caixas de diálogo de definição de parâmetros que reduzem o tempo de configuração que, juntamente com os blocos de funções *standard* em texto estruturado em conformidade com [43] ou numa linguagem em *ladder* [44] convencional, torna o desenvolvimento de programas para PLCs numa simples configuração de arrastar e largar *Drag & Drop* [45].

A Figura 24 apresenta a janela principal do CX-Programmer. Antes desta janela aparecer no ecrã, é necessário ao utilizador escolher qual o modelo de PLC que irá receber

o programa. Na parte superior desta janela podem ser encontrados os menus e as *toolbars* onde o utilizador pode aceder rapidamente aos contatos e blocos necessários à escrita do programa. À esquerda do *workspace* pode ser consultada toda a informação disponível sobre o modelo de PLC escolhido.

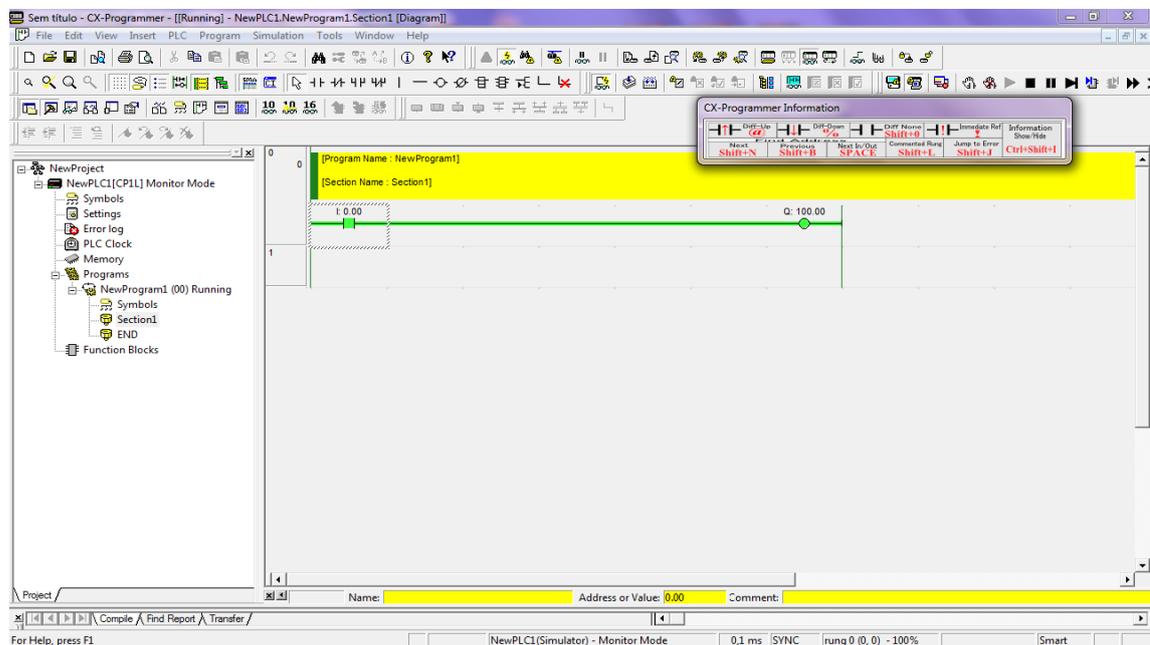


Figura 24 - Janela principal do CX-Programmer.

O CX-Programmer é uma ferramenta versátil que pode ser utilizada no desenvolvimento de simulações SIL, HIL e MIL. Após a edição do programa estar completa, este pode ser guardado e enviado para um PLC real (através de um dispositivo USB ou de um cabo de rede), estando pronto a ser utilizado. Caso o utilizador pretenda, pode também simular o programa num PLC virtual gerado pelo CX-Simulator, podendo verificar o seu comportamento.

### 3.2.3.2. CX-Simulator

O CX-Simulator é uma ferramenta de simulação que necessita que um programa de PLC esteja escrito no CX-Programmer para poder entrar em funcionamento.

Este *software* cria um PLC virtual, permitindo a avaliação da operação do programa, verificação do tempo de ciclo e redução do tempo da deteção e remoção de erros. Além disto, permite ainda utilizar vários métodos que permitem criar e reproduzir

entradas externas virtuais e executar operações de depuração eficazes que não possam ser executadas no PLC real [46].

O CX-Simulator é ativado através do CX-Programmer ou do CX-Designer e não apresenta nenhuma janela de funcionamento no ecrã do utilizador. No entanto é de fácil perceção de quando se encontra a correr um programa de PLC, pois desativa a maioria das funcionalidades dos programas enunciados anteriormente.

### 3.2.3.3. CX-Designer

O CX-Designer é um *software* HMI que permite ao utilizador criar, simular e desenvolver diferentes tipos de projetos de ecrã para os seus programas. Apresenta uma interface de utilizador totalmente personalizável com ícones para a maioria das funções e uma biblioteca de 1000 objetos funcionais padrão com gráficos e funções avançadas associados, tornando esta ferramenta muito fácil de utilizar mesmo por utilizadores inexperientes [47].

A Figura 25 apresenta a janela principal do CX-Designer, juntamente com pequenas descrições dos principais componentes do programa.

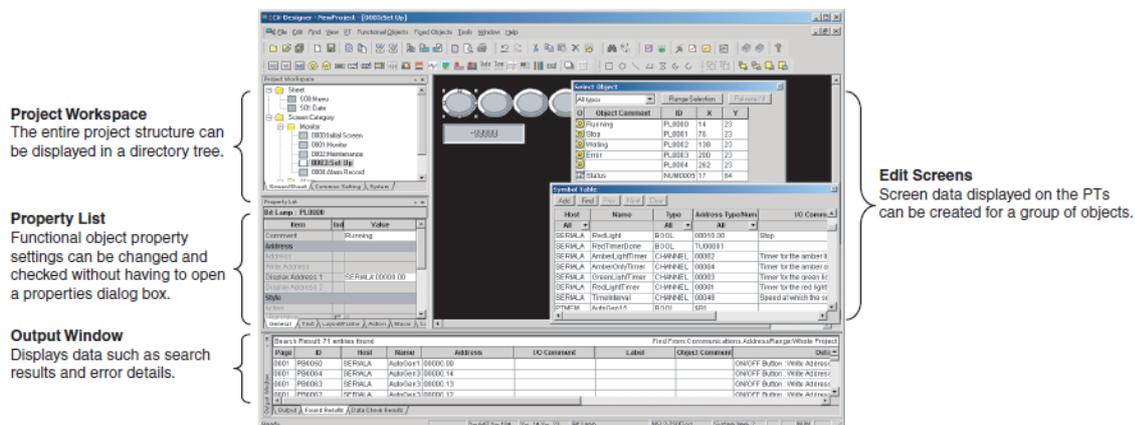


Figura 25 - Janela principal do CX-Designer e suas características [48].

O CX-Designer é uma ferramenta muito utilizada para simular programas para PLCs, sendo esta a sua principal funcionalidade. Através da ligação ao CX-Simulator, este *software* é capaz de interagir com o(s) programa(s) escrito(s) no CX-Programmer, podendo por isso ser utilizado no âmbito de simulações SiL e MiL.

### 3.3. Informação a reter do capítulo

A simulação é uma área cada vez mais importante na indústria. Devido ao grande desenvolvimento dos simuladores, é possível simular sistemas automatizados em condições muito próximas com as que existem na realidade, a custos muito mais reduzidos e com todo o conforto que uma secretária e um computador pessoal podem oferecer.

Existem quatro abordagens de simulação distintas, que podem ser utilizadas individualmente ou em conjunto, para a obtenção de melhores resultados. Nos testes de um sistema automatizado a abordagem de simulação deve ser criteriosamente escolhida, tendo em atenção os recursos financeiros e de logística disponíveis, bem como os riscos inerentes à sua simulação.

Apesar de existir uma lista enorme de simuladores que poderiam ser utilizados, para a realização deste trabalho foi escolhido o *software* CX-One da OMROM.

# Capítulo 4

## **METODOLOGIA DE CONSTRUÇÃO DAS BANCADAS DE SIMULAÇÃO**



## **4. METODOLOGIA DE CONSTRUÇÃO DAS BANCADAS DE SIMULAÇÃO**

Como se encontra referido no Capítulo 1 desta dissertação, um dos objetivos deste trabalho é a elaboração de uma metodologia padrão, que possa ser seguida por qualquer utilizador que tenha interesse em construir uma bancada de simulação.

Neste capítulo é proposta uma abordagem de construção dividida em seis passos distintos. De forma a poder ser seguida sem dificuldades, cada um dos passos é explicado através de exercícios ilustrativos, onde podem ser observadas todas as tarefas realizadas.

Com a função de servir de referência para trabalhos futuros nesta área, são também apresentados alguns resultados obtidos com a experimentação desta abordagem, tais como erros mais frequentes, sua origem e sugestões para a sua resolução.

### **4.1. Abordagem desenvolvida**

Para construir uma bancada de simulação a partir de um enunciado de um exercício, é necessário realizar algumas tarefas, tais como o estudo das características dos componentes do sistema, estudo do seu funcionamento pretendido e estudo da sua implementação no ambiente de simulação, etc.

Seguindo uma metodologia pré-definida, desde a análise do enunciado do problema até à sua simulação no ambiente de simulação, além de reduzir numa quantidade significativa o tempo desperdiçado com todo o trabalho referido no parágrafo anterior, tem também a vantagem de diminuir a probabilidade de surgirem dificuldades não previstas ao longo do processo de construção da bancada de simulação.

De seguida são apresentados os passos da metodologia desenvolvida com a experiência adquirida ao longo da elaboração deste trabalho. Para ser mais fácil seguir o raciocínio de cada passo, são apresentadas algumas ilustrações de um simples caso de estudo.

## Passo nº1 – Interpretação do enunciado do problema

O primeiro passo a ser realizado é a interpretação do comportamento pretendido para todos os componentes físicos a serem considerados na bancada.

Na Figura 26 encontra-se representado um exemplo pretendido para uma bancada de simulação.

Numa fábrica existe um posto de furação de peças automático, como o que se encontra representado na figura. Existem dois sensores, s1 e s2, que vão detetar a posição do berbequim em cima e em baixo, respetivamente. Assim que o botão START é premido pelo operador, a broca (B) entra em funcionamento, e o acionador desce. Ele permanecerá nessa posição até que o operador prima o botão STOP, que fará o acionador mover-se para cima e desligará a broca.

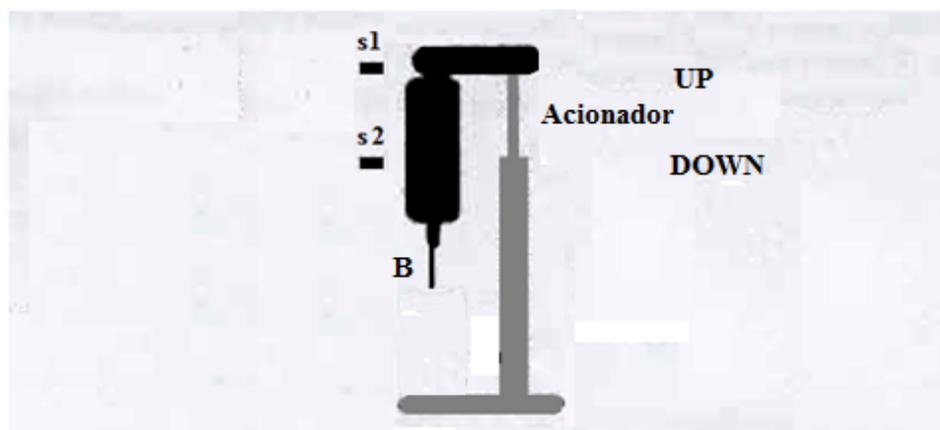


Figura 26 – Exemplo de um enunciado e representação de um exercício.

Analisando o que é pretendido para a bancada, conclui-se que o sistema é composto pelos seguintes componentes físicos:

- Broca;
- Berbequim;
- Acionador;
- Sensores s1 e s2.

Relativamente ao seu funcionamento, ao longo da simulação é esperado que a broca rode e desça quando seja premido o START. Por sua vez, ela deve subir e parar quando o botão STOP tiver sido premido.

## Passo nº2 – Construção de tabelas com as entradas e saídas do controlador

Como foi referido no Capítulo 1.2.2., no processo de simulação os modelos físicos e o controlador vão interagir através de variáveis Booleanas e não-Booleanas, simulando as trocas de sinais elétricos existentes entre o controlador real e a parte física real. A definição destas variáveis é importante para a modelação dos sistemas automatizados, pois:

- As entradas do controlador correspondem às saídas dos modelos da parte física;
- As saídas do controlador correspondem às entradas dos modelos da parte física.

A melhor forma de definir as variáveis a utilizar é a construção de tabelas de entradas e saídas do controlador, como as Tabelas 2 e 3 construídas para o exemplo apresentado. Além da informação da descrição de cada uma das variáveis, deve-se definir também os canais (*words*) e bits do controlador que irão ser utilizados para cada uma delas na simulação. Esta informação será importante na escrita do programa em *ladder* no CX-Programmer.

<b>Entradas</b>	<b>Descrição</b>	<b>Word &amp; Bit</b>
START	Botoneira de acionamento do sistema	0.00
STOP	Botoneira de paragem do sistema	0.01
s1	Berbequim na sua posição superior	0.02
s2	Berbequim na sua posição inferior	0.03

Tabela 2 – Entradas do controlador correspondentes às saídas da parte física.

<b>Saídas</b>	<b>Descrição</b>	<b>Word &amp; Bit</b>
A_DOWN	Movimento do berbequim para baixo	100.02
A_UP	Movimento do berbequim para cima	100.03
B	Acionamento da broca	100.04

Tabela 3 – Saídas do controlador correspondentes às entradas da parte física.

### Passo nº3 – Elaboração dos modelos da parte operativa

Depois de definidas as entradas e saídas do controlador e da parte física, já se encontram reunidas as condições necessárias para a construção dos modelos. Apesar de ao longo deste trabalho se ter utilizado o formalismo autômatos finitos temporizados e na maioria das vezes, a metodologia de modelação em modelos de grupos instanciáveis apresentada no Capítulo 2.3.2., outros formalismos e metodologias diferentes podem ser utilizadas para esta finalidade.

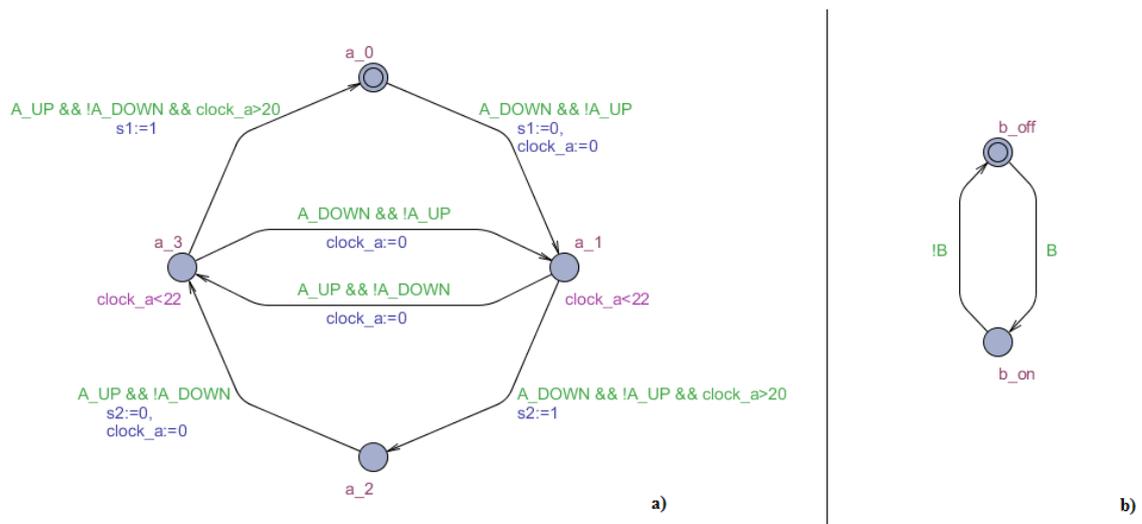


Figura 27 – Modelos construídos para o exemplo apresentado: a) modelo do grupo “atuador, berbequim e sensores”, b) modelo individual da broca.

### Passo nº4 – Conversão dos modelos para equações *ladder* e escrita no CX-Programmer

Para que a interação entre os programas do controlador e os modelos da parte física aconteça, é necessário que estejam escritos numa linguagem comum, que lhes permita estabelecer a comunicação. Desta forma, o formalismo autômatos finitos temporizados, utilizado para modelar os componentes pertencentes à parte operativa dos sistemas automatizados, foi convertido em linguagem *ladder* conforme a metodologia proposta em [49].

No CX-Programmer, as equações dos modelos obtidas devem-se encontrar sempre depois das equações do controlador. Este aspeto é fulcral para que não venham a

existir problemas ao longo da simulação. Na Figura 28 encontra-se representando o ambiente do CX-Programmer com as equações escritas do exemplo apresentado.

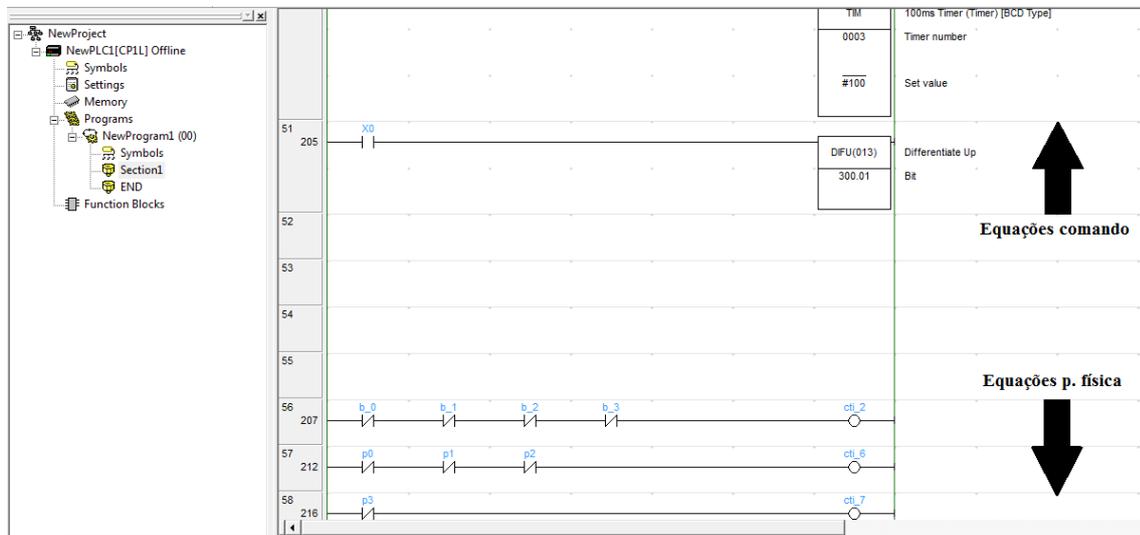


Figura 28 – Ambiente do CX-Programmer com as equações escritas.

## Passo nº5 – Construção do ambiente de simulação no CX-Designer

Ao longo do processo de simulação, é no ambiente do CX-Designer que o utilizador pode analisar e tirar as suas conclusões relativas aos seus programas de controlador. Através da observação do funcionamento da parte operativa, este deve poder avaliar o comportamento dos seus programas e ser capaz de apontar e corrigir as falhas que eles apresentam. Devido a estes motivos, a construção do ambiente de simulação deve ser cuidada, de modo a tentar evitar criar confusões e induzir o utilizador em erro.

Para auxiliar o desenho do sistema automatizado, o programa possui na sua base de dados uma biblioteca com mais de 1000 objetos pré-programados, tais como vários tipos de sensores, válvulas, lâmpadas, etc. Além disto, permite ainda a importação de figuras criadas e/ou editadas em *softwares* específicos de desenho, desde que estejam no formato *bitmap* (extensão “.bmp”). Desta forma podem ser utilizados desenhos mais realistas e fáceis de interpretar dos sistemas automatizados.

Na Figura 29 encontra-se representado o ambiente de simulação para o exercício proposto para exemplo.

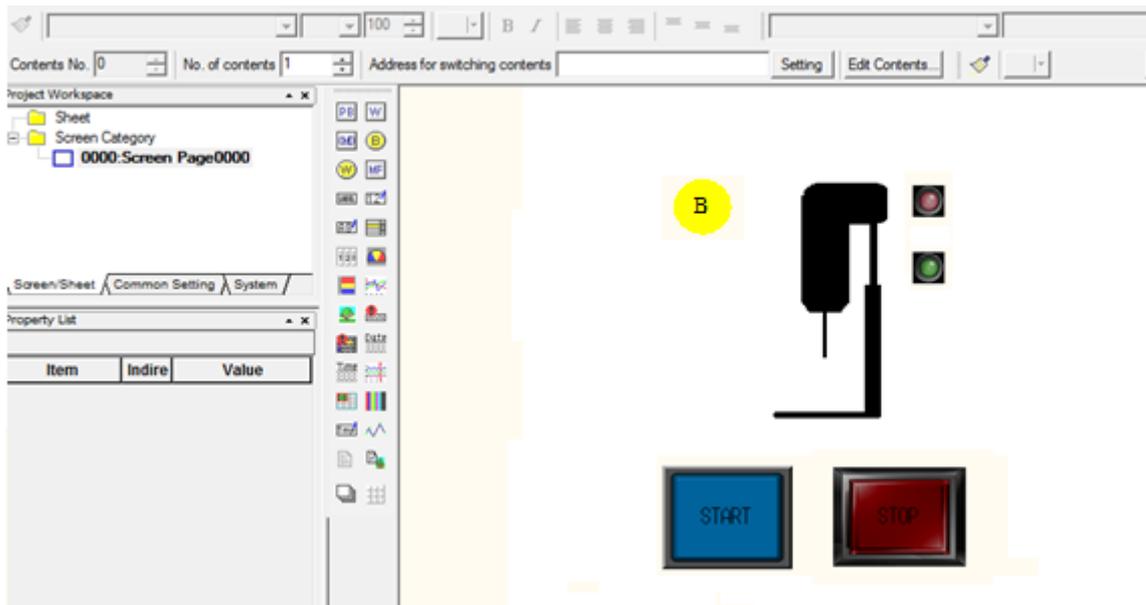


Figura 29 – Ambiente de simulação criado no CX-Designer para o exemplo proposto.

### Passo nº6 – Teste da bancada de simulação

A última tarefa resume-se em testar o funcionamento da bancada de simulação construída no passo anterior, e verificar se o seu comportamento é o esperado em cada um dos testes realizados. Para isto devem ser escritos programas de controlador que obriguem a parte física a fazer ações extremas, tais como por exemplo “o atuador mover para cima e para baixo em cada 10 segundos ao longo de 2 minutos” ou “atuar a broca quando o berbequim se encontra nas suas posições extremas, interrompendo o seu funcionamento quando este se encontra nas suas posições intermédias”. É graças a estes testes que podem ser detetados e corrigidos erros na construção, sendo no entanto difícil eliminá-los, a todos, em sistemas com maior grau de complexidade.

Na Figura 30 encontra-se representado um teste realizado no exercício utilizado como exemplo de aplicação.

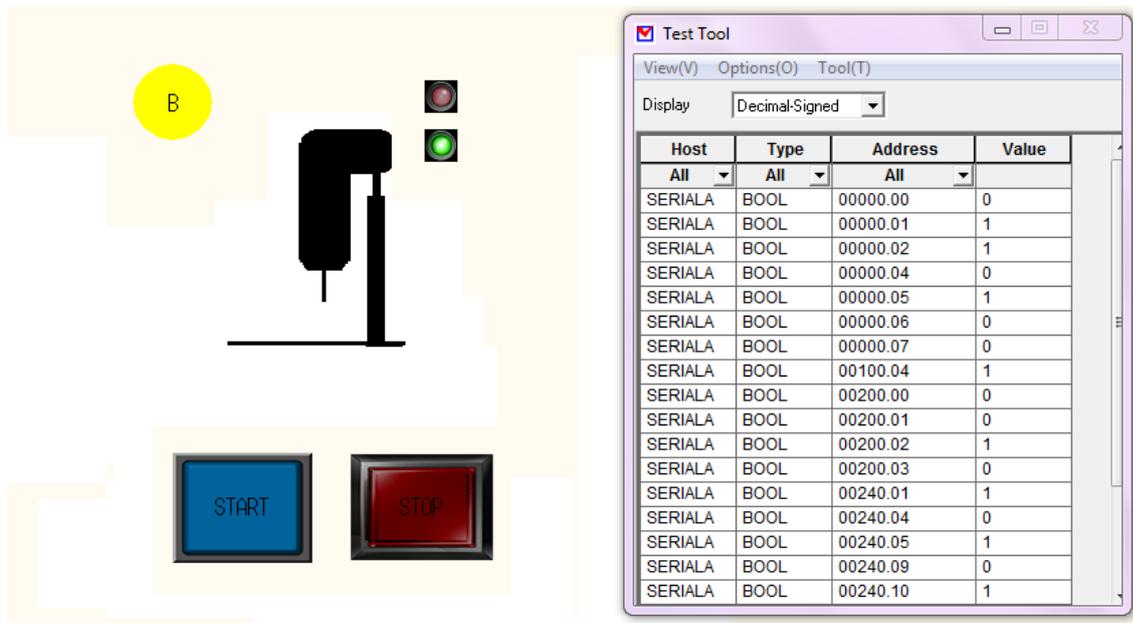


Figura 30 – Exemplo de um teste realizado.

## 4.2. Dificuldades encontradas

Ao longo do processo da construção de uma bancada de simulação, existe a probabilidade de surgirem problemas capazes de criar dificuldades, num ou em mais passos da metodologia apresentada no ponto anterior. Quando isto acontece, é então necessário despende tempo e recursos na procura de soluções, podendo esta tarefa tornar-se morosa e complexa. Devido a isto, é importante apontar os principais problemas encontrados ao longo da elaboração deste trabalho e apresentar algumas das suas soluções.

Um dos problemas que aparece com frequência surge da pouca precisão da informação com que, normalmente, se define uma bancada pretendida. Como foi referido anteriormente, é essencial haver a informação sobre as principais características dos componentes para que possam ser construídos modelos rigorosos, o mais próximos possível da realidade. Uma solução para este problema passa pela pesquisa em catálogos de fabricantes destes componentes, disponibilizados nas suas páginas de Internet. Desta forma é possível obter o acesso a esta informação, num curto intervalo de tempo.

Como foi demonstrado no Capítulo 2.2., cada formalismo de modelação possui as suas características específicas. Para evitar problemas futuros ao longo da simulação e

aproveitar as suas propriedades, antes de se passar para a construção dos modelos, deve ser feito um estudo aprofundado sobre o formalismo a ser utilizado. O mesmo deve ser feito em relação ao ambiente de simulação. Quanto maior for a quantidade de informação adquirida sobre as ferramentas que se vão utilizar, maior será a capacidade de aproveitar as suas características.

No entanto, é durante a execução da simulação e através da observação do comportamento do ambiente construído, que são detetados os problemas mais complicados de resolver, na forma de “erros de simulação”. Estes erros podem ter uma das seguintes origens:

- Na elaboração dos modelos;
- Na conversão de autómatos finitos temporizados para linguagem *ladder*;
- Na escrita das equações *ladder* no CX-Programmer;
- Erros do próprio *software* de simulação (muito improváveis, mas existem).

Devido às múltiplas possibilidades de origem dos erros, todo o processo (desde a análise do enunciado do exercício até à construção do ambiente de simulação) deve ser revisto com o objetivo de localizar a origem dos erros. Este processo pode ser longo e demorado, já que muitas vezes os erros não são detetados na primeira análise.

Ao longo da elaboração deste trabalho foram registados todos os erros detetados durante os testes realizados em cada uma das bancadas de simulação construídas. Como referência para outros utilizadores, na Figura 31 encontra-se representado um gráfico com a distribuição em percentagem, dos erros associados a cada uma das origens apresentadas anteriormente.

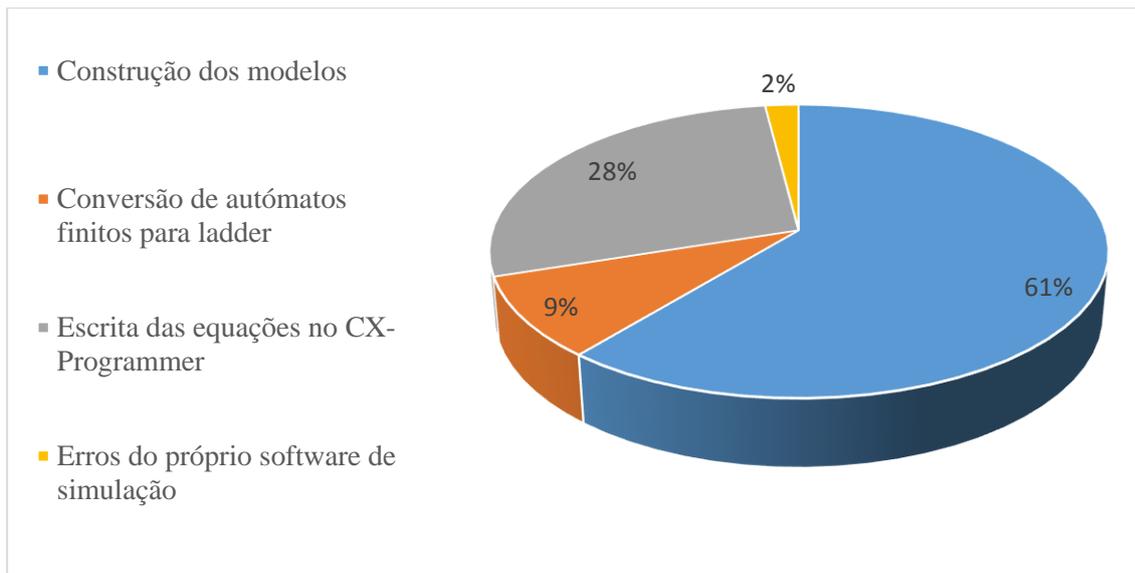


Figura 31 – Gráfico da proporção da origem dos erros surgidos na simulação ao longo do trabalho realizado.

Como é possível verificar no gráfico da Figura 31, a grande maioria (89%) dos erros detetados ao longo deste trabalho na simulação das bancadas de simulação, tiveram origem na construção dos modelos dos componentes físicos e na escrita das equações no CX-Programmer. No entanto, é importante realçar que os valores apresentados correspondem apenas aos erros detetados por um utilizador, devendo por isso ser considerados apenas como uma referência para outros utilizadores, em trabalhos futuros.

### 4.3. Informação a reter do capítulo

Para a construção de uma bancada de simulação a partir de um enunciado de um exercício deve ser seguida a seguinte lista de tarefas:

1. Interpretação do enunciado do problema;
2. Construção das tabelas com as entradas e saídas do controlador;
3. Elaboração dos modelos da parte operativa;
4. Conversão dos modelos para equações *ladder* e escrita no CX-Programmer;
5. Construção do ambiente de simulação no CX-Designer;
6. Teste da bancada de simulação.

Ao longo do processo apresentado em cima vão surgir dificuldades e erros ao longo da simulação. À medida que se vai adquirindo experiência, vão ser encontradas

mais facilmente soluções para a resolução de problemas. Por outro lado, para corrigir os erros de simulação, deve haver um foco nos modelos da parte operativa e nas equações do CX-Programmer, já que é aí que existe a maior probabilidade de ocorrerem.

# **Capítulo 5**

## **CASOS DE ESTUDO**



## 5. CASOS DE ESTUDO

Ao longo deste capítulo são apresentadas três bancadas de simulação com grau de complexidade crescente, desenvolvidas ao longo da dissertação.

Nestes exercícios encontra-se aplicada a abordagem explicada no Capítulo 4, bem como as metodologias de modelação da parte operativa referidos no Capítulo 2. São também apresentadas as condições necessárias para estas bancadas poderem ser utilizadas no âmbito do ensino na automação.

### 5.1. Caso de estudo nº1 – Barreira automática

O sistema automatizado representado na Figura 32 controla a entrada de veículos para um parque de estacionamento. Ele é constituído por uma barreira acionada por um motor com dois sentidos de rotação: movimento de subida (MIS) e movimento de descida (MID). Além disso, existe também um conjunto de quatro sensores:

- Dois para detetar a presença de viatura quando se encontra na entrada da barreira (se1) e quando a atravessa (se2);
- Dois para a deteção da posição da barreira, sendo um para detetar quando se encontra totalmente subida (sbs) e outro para quando estiver totalmente descida (sbd).

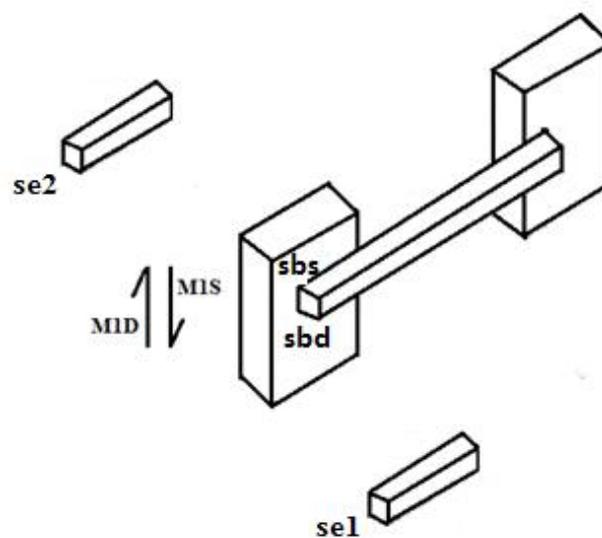


Figura 32 – Barreira automática e sensores constituintes do sistema a modelar.

Quando se aproxima uma viatura (se1 atuado), a barreira deve subir e deve permanecer levantada até que não seja detetada viatura por se2. Depois disto, a barreira deve descer, mas se porventura, nalgum instante durante a descida, o sensor se1 detetar a presença de uma nova viatura, a barreira deve subir imediatamente para não a danificar.

Com a finalidade de tratamento estatístico, no final de cada dia deve ser possível saber quantos carros entraram no parque. Para este efeito devem ser contabilizadas as vezes que o sensor se2 liga e desliga.

### **5.1.1. Modelação do sistema automatizado**

Como foi referido no enunciado do exercício, existem no total seis componentes físicos (motor, barreira e sensores) que necessitam de ser modelados para simular o comportamento do sistema. Analisando o problema, tiram-se duas conclusões importantes necessárias para a construção dos modelos:

1. Os sensores sbs e sbd são dependentes do posicionamento da barreira. Eles ligam quando a barreira se encontra em cima e em baixo, respetivamente. Por outro lado, a barreira move-se para cima e para baixo dependendo do movimento do motor;
2. Os sensores se1 e se2 vão depender exclusivamente das viaturas, atuando apenas quando estas atravessam as posições onde se encontram colocados. Desta forma afirma-se que são totalmente independentes do comportamento dos restantes componentes físicos do sistema.

Atendendo a estes dois pontos pode-se concluir que para modelar o sistema automatizado vão ser necessários dois modelos. O primeiro terá de modelar o comportamento do grupo composto pelo motor, barreira e sensores sbs e sbd. O segundo modelo terá a função de modelar o comportamento das viaturas e dos sensores se1 e se2.

### 5.1.1.1. Tabelas de entradas e saídas do controlador

Entradas	Descrição	Word & Bit
se1	Presença de viatura à entrada da barreira	0.02
sbd	Barreira na posição superior	0.03
sbs	Barreira na posição inferior	0.04
se2	Presença de viatura à saída da barreira	0.05

Tabela 4 – Entradas do controlador/saídas da parte operativa.

Saídas	Descrição	Word & Bit
M1S	Movimento de subida da barreira	100.00
M1D	Movimento da descida da barreira	100.01

Tabela 5 – Saídas do controlador/entradas da parte operativa

### 5.1.1.2. Modelo da barreira

Na Figura 33 encontra-se representado o modelo do grupo elaborado, constituído pelo motor, barreira e sensores sbs e sbd.

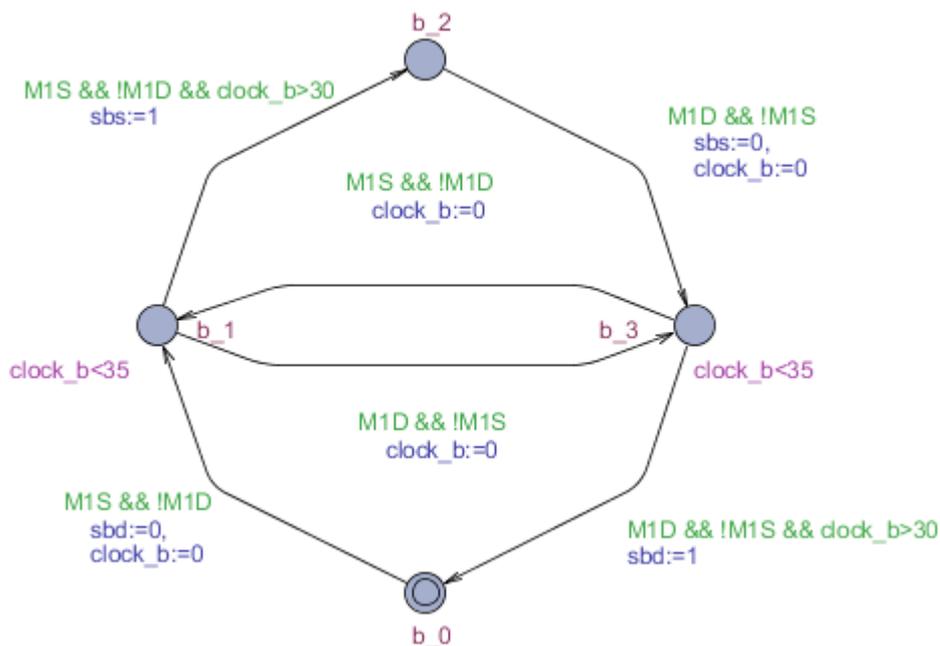


Figura 33 – Modelo do grupo “motor-barreira-sensores sbs e sbd”.

<b>Estados</b>	<b>Descrição</b>
b_0	Barreira na posição inferior (estado inicial)
b_1	Barreira a levantar
b_2	Barreira na posição superior
b_3	Barreira a descer

Tabela 6 – Descrição de cada um dos estados do modelo do grupo “motor-barreira-sensores sbs e sbd”.

Como pode ser visto na figura anterior, o modelo é composto por quatro estados. Cada um deles diz respeito a uma posição distinta da barreira, como se encontra representado na tabela 6.

As transições entre os estados do modelo possuem como guardas as ordens emitidas pelo controlador ao motor e relógios virtuais. Estes são necessários para simular o tempo necessário para que a barreira, localizada numa posição intermédia, suba ou desça totalmente. Além disto, possuem ainda atribuição de variáveis aos sensores da barreira, sbs e sbd. São estas que vão controlar o comportamento dos sensores, atribuindo-lhes os valores “1” e “0” caso devam ligar ou desligar, respetivamente.

### 5.1.1.3. Modelo das viaturas

O segundo modelo elaborado foi o das viaturas. Este é totalmente independente do controlador, visto que não existe forma de controlar o número e a periodicidade das viaturas que chegam ao pé da barreira para a atravessar. Por este motivo, foi construído um modelo observador, com o objetivo de ser capaz de reproduzir os seguintes comportamentos:

- As viaturas aparecem aleatoriamente ao pé da barreira e sem periodicidade definida. Cada vez que uma viatura nova aparece, se1 deve ligar;
- Enquanto a barreira sobe, o “condutor” da viatura tem a possibilidade de decidir se vai ultrapassá-la ou inverter o sentido sem a trespassar. A decisão final é apenas tomada na altura em que a barreira se encontra completamente erguida. Caso a ultrapasse, existe um momento em que se2 liga, desligando mal a viatura deixe de se encontrar na posição ao alcance do sensor.

Atendendo aos comportamentos acima referidos, na Figura 34 encontra-se representado o modelo observador.

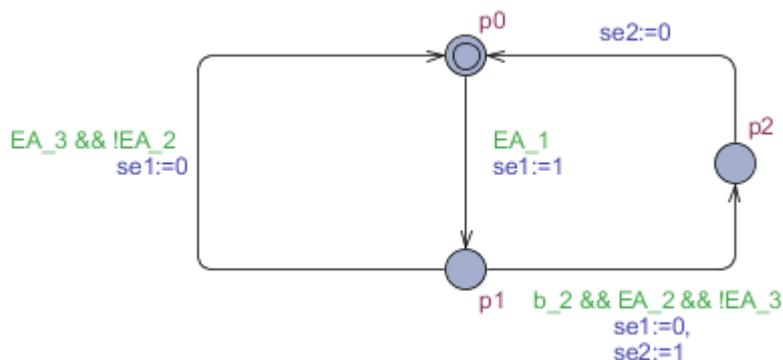


Figura 34 – Modelo observador das viaturas.

Estados	Descrição
p0	Sem viatura no sistema (estado inicial)
p1	Viatura no sensor se1, antes da barreira
p2	Viatura no sensor se2, depois da barreira

Tabela 7 – Descrição de cada um dos estados do modelo observador das viaturas.

Apesar do modelo ser de reduzidas dimensões, este simula corretamente o comportamento da viatura referido anteriormente. Como é possível observar na figura, à exceção da transição de “p2” para “p0”, todas as outras possuem guardas com eventos aleatórios. Estes são necessários para simular:

- A aparição de viaturas na barreira (EA\_1);
- A decisão de atravessar a barreira (EA\_2);
- A decisão de não atravessar a barreira (EA\_3).

Além disto, o modelo tem a função de ligar e desligar os sensores se1 e se2. Estes encontram-se ligados apenas nos estados “p1” e “p2”, respetivamente.

## 5.1.2. Simulação

### 5.1.2.1. Alterações efetuadas nos modelos

Antes de se proceder à simulação, foi necessário fazer alterações ao modelo construído para simular as viaturas. O modelo apresentado na Figura 34 possui um número reduzido de estados (apenas três), sendo um pouco reduzido para representar num ambiente de simulação. Visto cada estado corresponder a uma posição diferente de uma viatura no ambiente de simulação, mais estados devem ser adicionados ao modelo para

poderem “aparecer” mais viaturas no ecrã, e poder existir a ideia de movimento destas. Desta forma será construído um ambiente virtual no CX-Designer mais agradável e perceptível ao estudante, promovendo uma melhor e mais rápida aprendizagem evitando mal entendidos e confusões ao longo da simulação dos programas do controlador.

Na Figura 35 encontra-se representado o novo modelo construído para simular o comportamento das viaturas, juntamente com pequenos comentários para ser mais fácil de entender. Como é possível observar, este novo modelo é uma variante do modelo original, onde foram adicionados mais estados que servirão de posições ocupadas pelas viaturas ao longo da simulação.

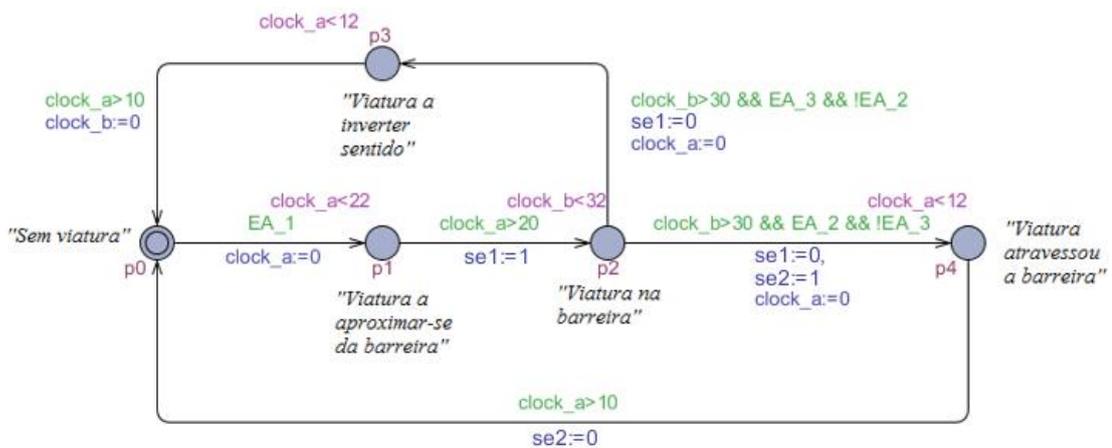


Figura 35 – Modelo das viaturas utilizado na simulação.

Além da adição de estados, foram também adicionados relógios em todas as transições (exceto na transição de “p0” para “p1”). Estes relógios servem para o aluno ter tempo de observar as viaturas “estagnadas” no ecrã do seu computador antes de avançarem para os estados seguintes. Assim consegue ter uma melhor perceção do que está a acontecer no ambiente virtual e analisar da melhor forma o desempenho dos seus programas do controlador.

### 5.1.2.2. Ambiente de simulação

Os modelos apresentados anteriormente foram traduzidos para equações da linguagem *ladder* e transcritos para o CX-Programmer, como se encontra representado na figura 36. Por uma questão de espaço, é apenas representada na figura uma parte das equações.

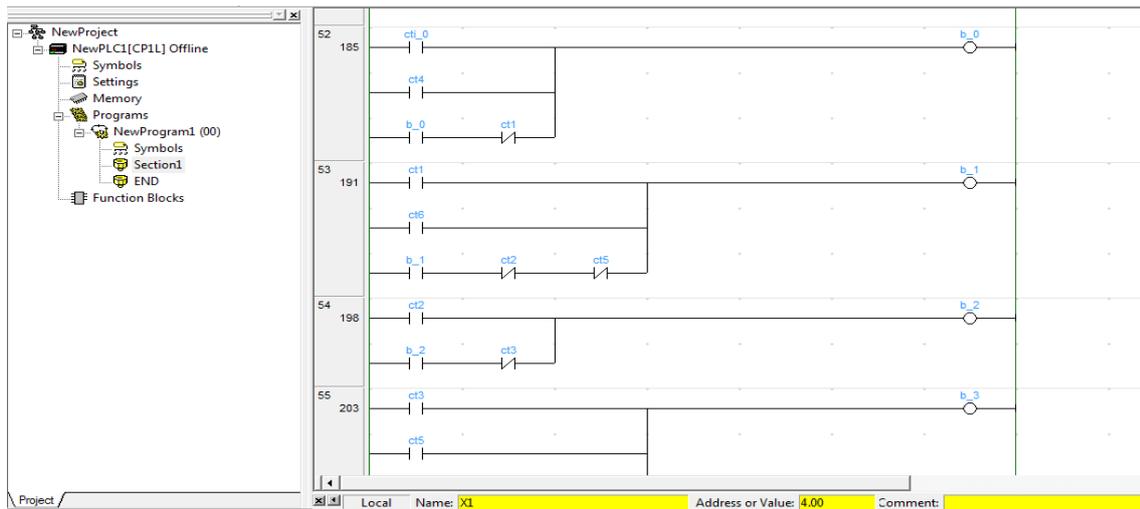


Figura 36 – Representação parcial das equações ladder dos modelos elaborados.

No processo de simulação, as equações representadas na figura anterior estarão ligadas pelo CX-Simulator ao ambiente virtual construído no CX-Designer, representado na figura 37.



Figura 37 – Ambiente de simulação criado no CX-Designer.

Procedendo à simulação, o ambiente virtual vai-se comportar como ordenado pelo programa do controlador que tiver sido escrito no mesmo ficheiro do CX-Programmer utilizado para escrever as equações da parte operativa. Nas Figuras 38 e 39 encontram-se representadas duas imagens do ambiente de simulação virtual CX\_Designer recolhidas ao longo da simulação, simulando os dois comportamentos distintos que os carros podem ter.



Figura 38 – Ambiente do CX-Designer ao longo da simulação: Viatura ultrapassou a barreira.



Figura 39 - Ambiente do CX-Designer ao longo da simulação: Viatura não ultrapassou a barreira e inverteu o sentido de marcha.

## 5.2. Caso de estudo nº2 – Misturadora de solução química

A Figura 40 representa uma instalação para preparação de uma solução química, na qual intervêm o produto A, o produto B e o solvente S.

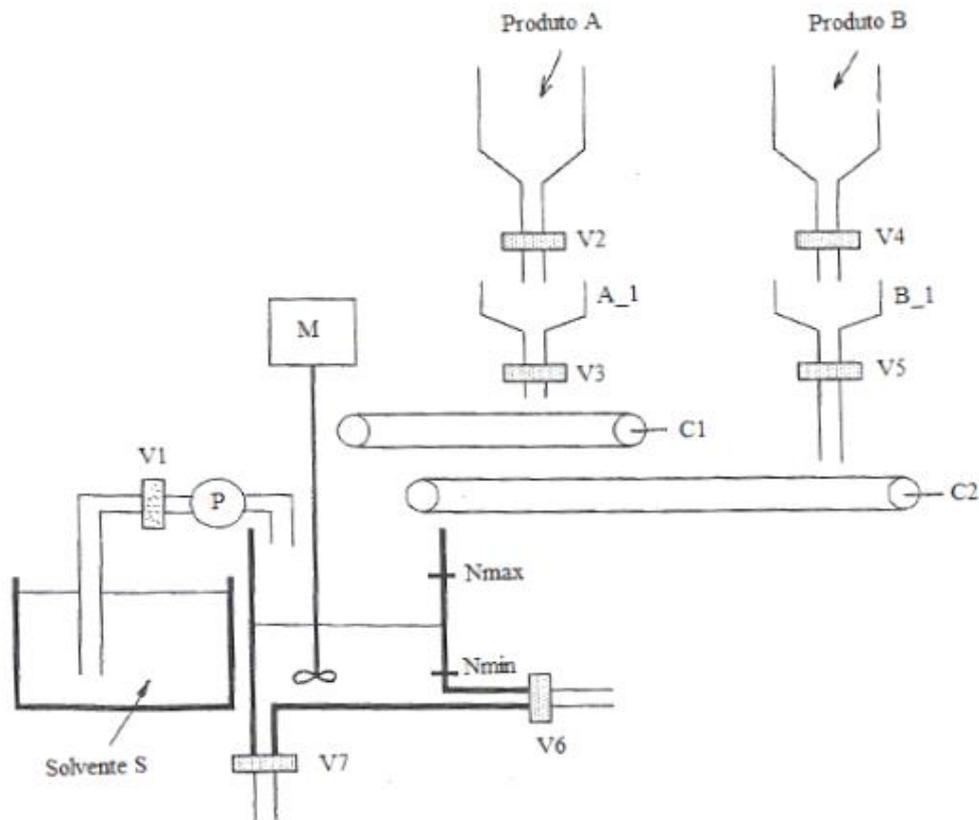


Figura 40 – Instalação de preparação da solução química.

O funcionamento da instalação é o seguinte:

- Se o nível da solução, na cuba misturadora, for inferior ao mínimo (Nmin), e o operador acionar o botão de arranque (START), as válvulas V1, V2 e V4 abrem-se e o motor (P), da bomba de aspiração do solvente, entra em funcionamento;
- Quando uma determinada quantidade dos produtos A e B tiver sido lançada sobre as básculas A\_1 e B\_1, respetivamente, as válvulas V2 e V4 fecham-se. Entretanto quando se atingir o nível máximo (Nmax) de solvente na cuba misturadora, o motor (P) da bomba é desligado, assim como é também fechada a válvula V1;
- Quando as três condições referidas no ponto anterior se verificarem, o motor M da hélice misturadora entra em funcionamento, os motores dos tapetes

transportadores recebem as ordens C1\_AV e C2\_AV, iniciando também o seu funcionamento e, simultaneamente, abrem-se as válvulas V3 e V5;

- Ao fim de 10 segundos as válvulas V3 e V5 são fechadas, assim como ambos os tapetes rolantes param em simultâneo;
- O motor M, da hélice misturadora, mantém-se em funcionamento por mais 20 segundos, ao fim dos quais parará, abrindo-se a válvula V6 para enviar a solução para o posto de utilização, mantendo-se aberta ao longo de 15 segundos;
- Se durante esses 20 segundos o operador considerar que a solução obtida não é satisfatória, acionará o botão REG, que provocará a paragem do motor M e a abertura da válvula V7, escoando-se a solução para o contentor apropriado;
- Quando o nível mínimo (Nmin) de solução na cuba for atingido, a válvula V6 ou V7, conforme a evolução havida, fechará e retornar-se-á o estado inicial da máquina.

É admitido que os produtos A e B encontram-se sempre disponíveis nos seus postos de armazenamento e que todas as válvulas são monoestáveis de retorno por mola.

### **5.2.1. Modelação do sistema automatizado**

O sistema apresentado neste caso de estudo é constituído por um número bastante significativo de componentes físicos, como se encontra referido no enunciado e representado na Figura 40. Além disto, existe a particularidade de existirem “objetos sólidos” (produtos A e B) e “objetos líquidos” (solvente e solução) simultaneamente, que tal como os restantes componentes físicos, necessitam de ser modelados para ser possível simular corretamente o comportamento do sistema.

Analisando o enunciado do exercício é possível estabelecer algumas conclusões importantes, necessárias para a construção dos modelos:

1. Os produtos A e B vão-se “mover” no sistema em direção ao tanque onde se encontra a solução, à medida que são atuadas por ordem duas sequências de componentes físicos. O produto A para avançar necessitará da sequência: “V2 - V3 - C1” e o produto B da sequência: “V4 - V5 - C2”;

2. Os sensores A\_1 e B\_1 dependem exclusivamente das quantidades dos produtos A e B existentes nas balanças, ao final de um tempo. Eles irão “ligar” assim que as ordens V2 e V4 sejam mantidas ao final de um tempo, respetivamente;
3. Para o solvente S se mover para o tanque da solução, será necessário a válvula V1 e a bomba P estarem atuados, simultaneamente. Só com esta condição é que o líquido percorre a canalização em direção ao tanque onde será feita a mistura;
4. Os sensores Nmin e Nmax dependem exclusivamente do volume de solução presente no tanque de mistura. Este depende de três fatores distintos: quantidade do produto A, quantidade do produto B e volume de solvente S que é inserido no tanque de mistura;

Como já foi referido anteriormente, é imperativo que os modelos sejam o mais compactos possível, para que a sua verificação formal seja mais rápida e versátil. Por este motivo, a maioria dos componentes físicos foram agrupados juntamente com os “objetos” com que interagem ao longo do processo:

- O motor C1 e as válvulas V2 e V3 foram agrupados ao modelo do produto A;
- O motor C2 e as válvulas V4 e V5 foram agrupados ao modelo do produto B;
- A válvula V1 e a bomba P foram agrupadas ao modelo do solvente S;
- As válvulas V6 e V7 foram agrupadas ao modelo da solução.

Desta forma, foram elaborados quatro modelos de grupo distintos, em detrimento dos modelos individuais de cada um dos constituintes do sistema. No entanto, para o motor M teve de ser construído o seu modelo individual. Visto não poder ser inserido em nenhum dos grupos referidos, foi obrigado a ser criada uma exceção à regra estabelecida.

#### 5.2.1.1. Tabelas de entradas e saídas do controlador

Entradas	Descrição	Word & Bit
START	Botão de acionamento do sistema	0.00
REG	Botão de paragem do motor M e acionamento da válvula 7	0.01
Nmin	Nível mínimo de solução no tanque	0.02

Nmax	Nível máximo de solução no tanque	0.03
A_1	Peso correto do produto A	0.04
B_1	Peso correto do produto B	0.05

Tabela 8 - Entradas do controlador/saídas da parte operativa.

Saídas	Descrição	Word & Bit
V1	Acionamento da válvula V1	100.00
V2	Acionamento da válvula V2	100.01
V3	Acionamento da válvula V3	100.02
V4	Acionamento da válvula V4	100.03
V5	Acionamento da válvula V5	100.04
V6	Acionamento da válvula V6	100.05
V7	Acionamento da válvula V7	100.06
P	Acionamento da bomba hidráulica P	100.07
M	Acionamento do motor da misturadora M	101.01
C1_AV	Movimento do tapete rolante C1 para a frente	101.02
C1_REC	Movimento do tapete rolante C1 para trás	101.03
C2_AV	Movimento do tapete rolante C2 para a frente	101.04
C2_REC	Movimento do tapete rolante C2 para trás	101.05

Tabela 9 – Saídas do controlador/entradas da parte operativa.

### 5.2.1.2. Modelo do produto A

Como foi referido no ponto anterior, o modelo do produto A é um modelo observador que, além de ter a função de modelar o comportamento deste produto, têm também de modelar o comportamento do motor C1 e das válvulas V2 e V3. Na Figura 41 encontra-se representado o modelo elaborado.

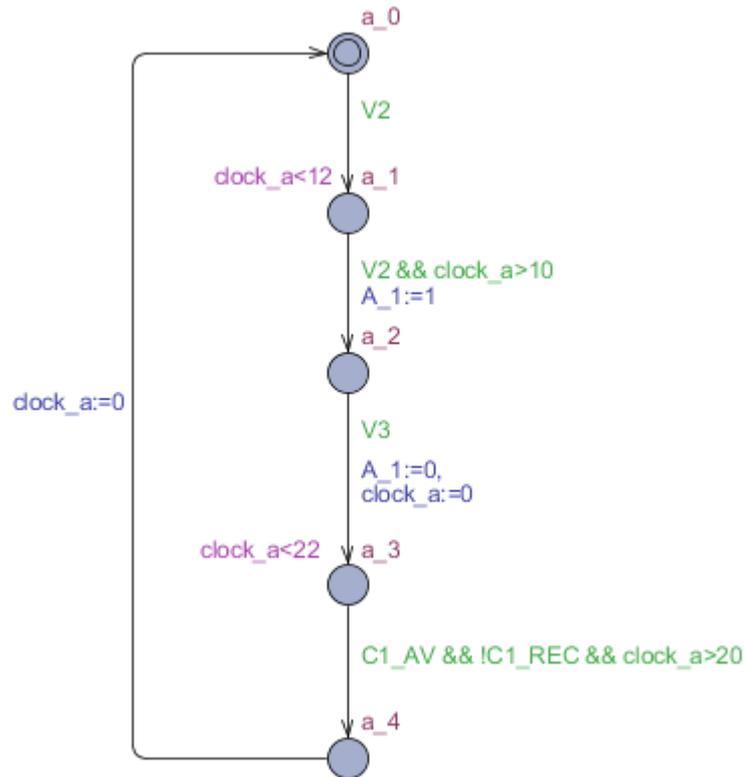


Figura 41 – Modelo do produto A.

Estados	Descrição
a_0	Produto A armazenado (estado inicial)
a_1	Produto A a cair para a balança
a_2	Quantidade específica do produto A medida na balança
a_3	Produto A no tapete rolante
a_4	Produto A no tanque de mistura

Tabela 10 – Descrição de cada um dos estados do modelo do produto A.

O modelo possui cinco estados, cada um deles a representar uma posição específica do produto A entre a sua posição inicial (armazenamento) e a sua final (tanque de mistura), como representado na tabela 10. A passagem de um estado para o seguinte

ocorre sempre que a guarda da transição que o leva a este seja verdadeira, sendo assim possível observar o comportamento dos componentes físicos agrupados a este modelo.

É importante lembrar que no enunciado do problema é referido que se encontra sempre disponível produto no local de armazenamento. Esta condição facilita a modelação, visto que deixa de ser necessário recorrer a eventos aleatórios para simular o aparecimento do produto no sistema.

Tal como é possível observar na figura, este modelo é responsável por atuar o sensor A\_1, correspondente ao peso do produto medido na balança. Este é atuado através da atribuição de variável presente na transição entre os estados “a\_1” e “a\_2”.

### **5.2.1.3. Modelo do produto B**

O modelo do produto B encontra-se representado na Figura 42. Este é um modelo observador construído para modelar o comportamento do produto B ao longo do processo de simulação, mas tratando-se do modelo de um grupo, tem também a função de permitir observar os comportamentos dos componentes físicos que se encontram a ele associados (neste caso o motor C2 e as válvulas V4 e V5).

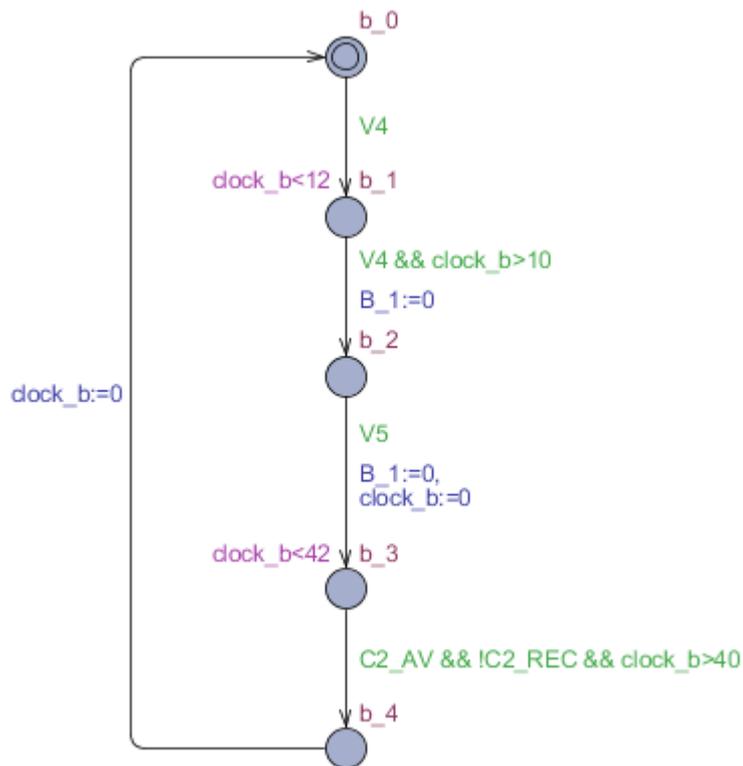


Figura 42 – Modelo do produto B.

Estados	Descrição
b_0	Produto B armazenado (estado inicial)
b_1	Produto B a cair para a balança
b_2	Quantidade específica do produto B medida na balança
b_3	Produto B no tapete rolante
b_4	Produto B no tanque de mistura

Tabela 11 – Descrição de cada um dos estados do modelo do produto B.

Como é possível observar na figura anterior, este é um modelo muito semelhante ao modelo do produto A, com a exceção de possuir guardas com condições distintas. Possui também cinco estados, cada um deles a representar uma posição diferente do produto no sistema conforme descrito na tabela 11. O sensor da balança B\_1 é ativado através da atribuição de variável localizada na transição que liga os estados “b\_1” e “b\_2”.

#### 5.2.1.4. Modelo do Solvente S

Ao contrário dos produtos A e B, o solvente S é um “objeto” que se encontra no estado líquido, podendo ser água, diluente ou outro tipo de fluído. Apesar desta diferença de estados, a modelação é feita da mesma forma, havendo apenas a diferença de os estados dos modelos representarem ações que o líquido está a sofrer ou a realizar (tais como “a encher” ou “a esvaziar”), em vez de representar posições estáticas como no caso dos sólidos. Esta distinção é importante, pois permite a construção de modelos mais simples, com menor número de estados e transições.

Na Figura 43 encontra-se representado o modelo elaborado para observar o comportamento do solvente.

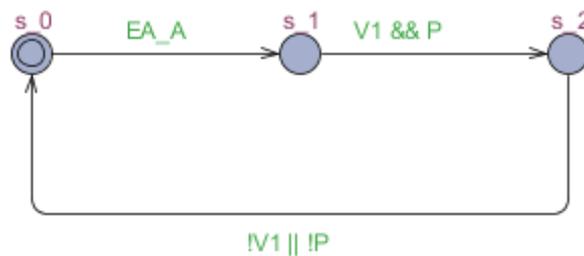


Figura 43 – Modelo do solvente S.

Estados	Descrição
s_0	Tanque de armazenamento do solvente S vazio – Solvente não existente no sistema (estado inicial)
s_1	Tanque do armazenamento do Solvente S cheio
s_2	Solvente S a passar para o tanque de mistura

Tabela 12 – Descrição de cada um dos estados do modelo do solvente S.

Como é possível observar na figura, o modelo do solvente é constituído por três estados distintos. Ao contrário dos produtos A e B, o solvente S depois de utilizado necessita de ser repostado no sistema para poder ser utilizado novamente. Devido a isto, foi necessário criar o evento aleatório *EA\_A*, presente na guarda da transição entre o estado inicial “s\_0” e o estado “s\_1”, para simular a ação de um operador ou máquina a repor o *stock* do fluído. Assim que se encontrar disponível (estado “s\_1” ativo), o solvente S irá correr para o tanque de mistura até que sejam desligadas a válvula V1 ou a bomba P.

### 5.2.1.5. Modelo da solução química

Para simular o comportamento da solução química (produto final do sistema), foi criado o modelo representado na Figura 44. Tal como no caso do solvente S, a solução é um produto no estado líquido, apresentando por este motivo as mesmas características de modelação referidas no modelo anterior.

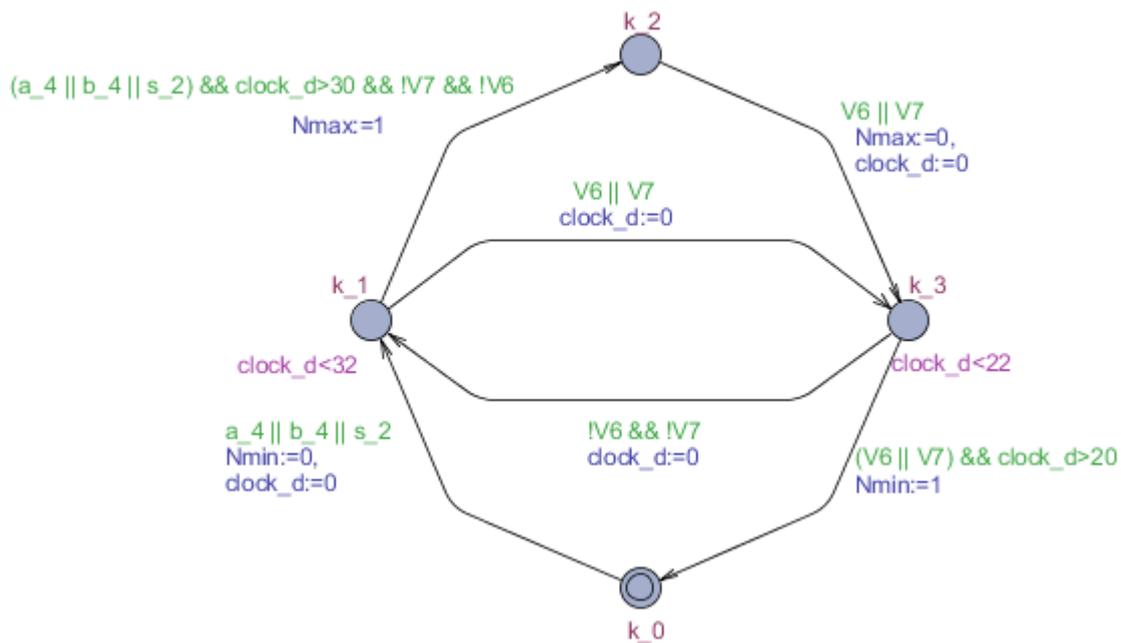


Figura 44 – Modelo da solução química.

Estados	Descrição
k_0	Tanque de mistura vazio – Solução não existente no sistema (estado inicial)
k_1	Tanque da solução a encher
k_2	Tanque da solução cheio
k_3	Tanque da solução ao esvaziar

Tabela 13 – Descrição de cada um dos estados do modelo da solução.

O modelo é constituído por quatro estados, cada um correspondente a uma posição específica do líquido da solução conforme representado na tabela 13. No estado inicial “k\_0”, quando não existe solução no tanque e este se encontra vazio, é atuado o sensor Nmin. Por outro lado, quando o modelo de encontra no estado “k\_2”, correspondente ao tanque estar cheio de solução química, é atuado o sensor Nmax.

O tanque enche conforme lhe são adicionados os produtos A e B e o solvente S, sem haver ordem estabelecida. Desta forma podem ser adicionadas diferentes quantidades de produtos ou solvente, ao longo da simulação. Para o esvaziar, basta abrir a válvula V6 ou V7, voltando à posição inicial ao final de um tempo definido no modelo.

## **5.2.2. Simulação**

### **5.2.2.1. Alterações efetuadas nos modelos**

Antes de se partir para a simulação, algumas alterações foram realizadas nos modelos dos produtos A e B para permitir ao utilizador obter o máximo rendimento desta experiência. Apesar dos modelos apresentados anteriormente simularem corretamente o comportamento do sistema, quando fosse realizada a simulação, o ambiente de simulação do CX\_Designer tornar-se-ia confuso para o utilizador, pois não iria conseguir visualizar os produtos a serem transportados entre as posições iniciais de armazenamento até ao tanque onde ocorre a mistura da solução, devido ao reduzido número de estados que os modelos possuem.

Para resolver este problema foram utilizados novos modelos para os produtos A e B, representados na Figura 45.

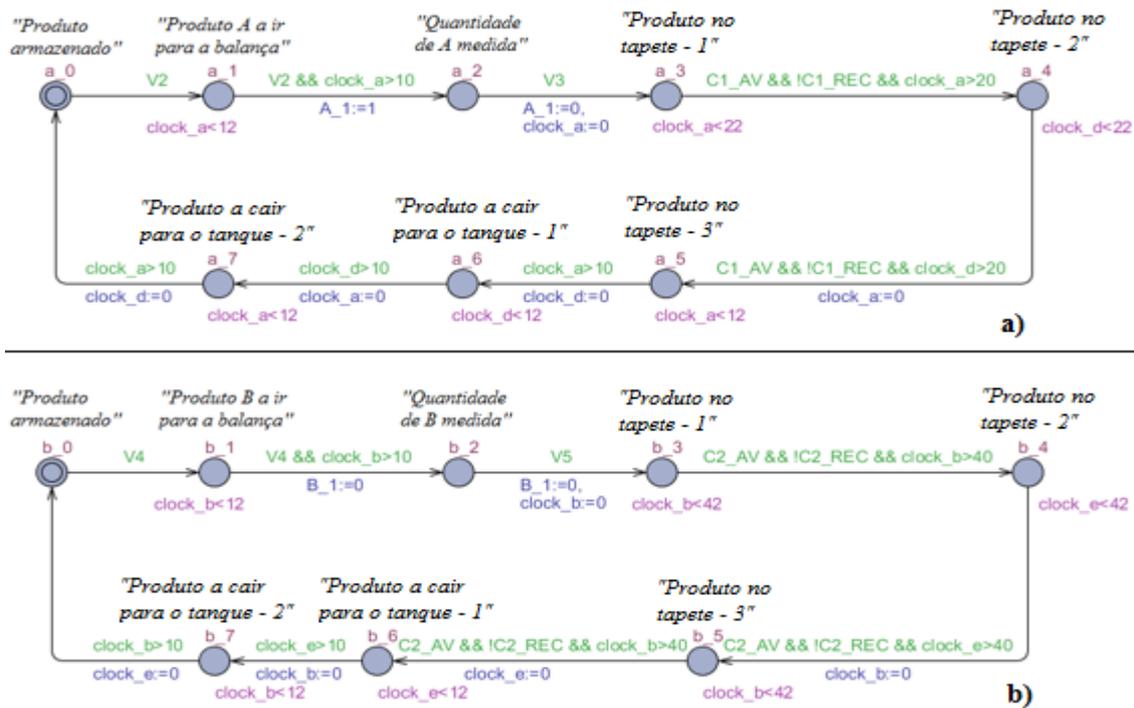


Figura 45 – Modelos reconstruídos através das alterações para a simulação: a) Produto A, b) Produto B.

Como é possível observar na Figura 45, foram acrescentados mais estados aos novos modelos, em comparação aos modelos dos produtos A e B das Figuras 41 e 42. Estes estados são importantes para permitir que seja criado o “movimento” destes produtos no ambiente de simulação do CX-Designer. Desta forma o utilizador pode acompanhar a simulação dos seus programas de controlador e observar a movimentação dos produtos no sistema.

Como consequência desta alteração realizada, outra alteração teve de ser efetuada num dos modelos observadores, nomeadamente no modelo correspondente ao da solução. Visto os estados responsáveis pela queda dos produtos A e B terem agora designações diferentes das anteriores, foi necessário corrigir as transições do modelo da solução responsáveis pelo enchimento do tanque. Assim, onde anteriormente existia “a\_4” e “b\_4” passa a ser “a\_7” e “b\_7”. O modelo em questão encontra-se representado na Figura 46, já com as transições devidamente corrigidas.

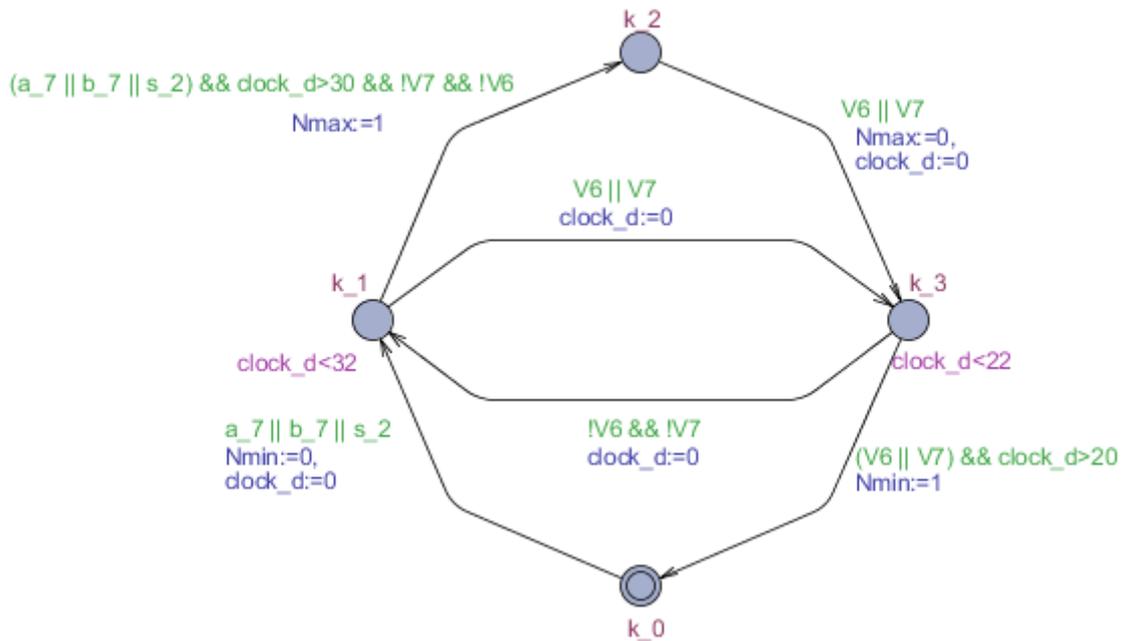


Figura 46 – Modelo da solução corrigido.

### 5.2.2.2. Ambiente de simulação

Os modelos foram traduzidos para equações da linguagem *ladder* e transcritos para o CX-Programmer, como se encontra representado na Figura 47. Por uma questão de espaço, é apenas apresentada uma figura de uma parte das equações obtidas.

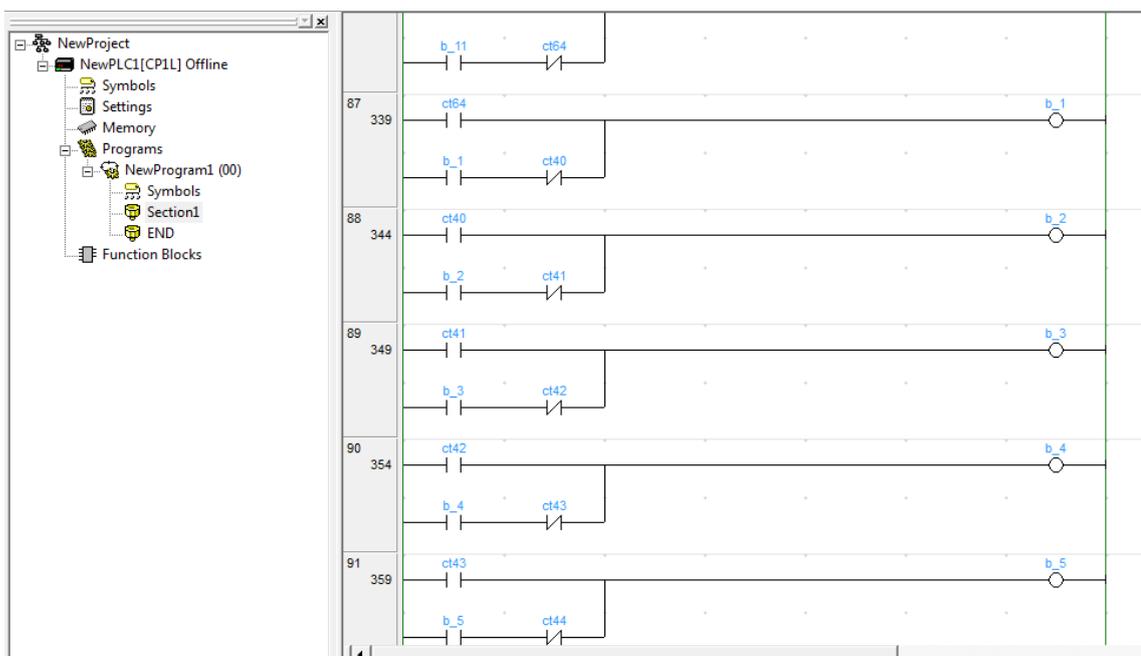


Figura 47 - Representação parcial das equações ladder dos modelos construídos.

No processo de simulação, as equações representadas na figura anterior estarão ligadas pelo CX-Simulator ao ambiente virtual construído no CX-Designer, representado na Figura 48.

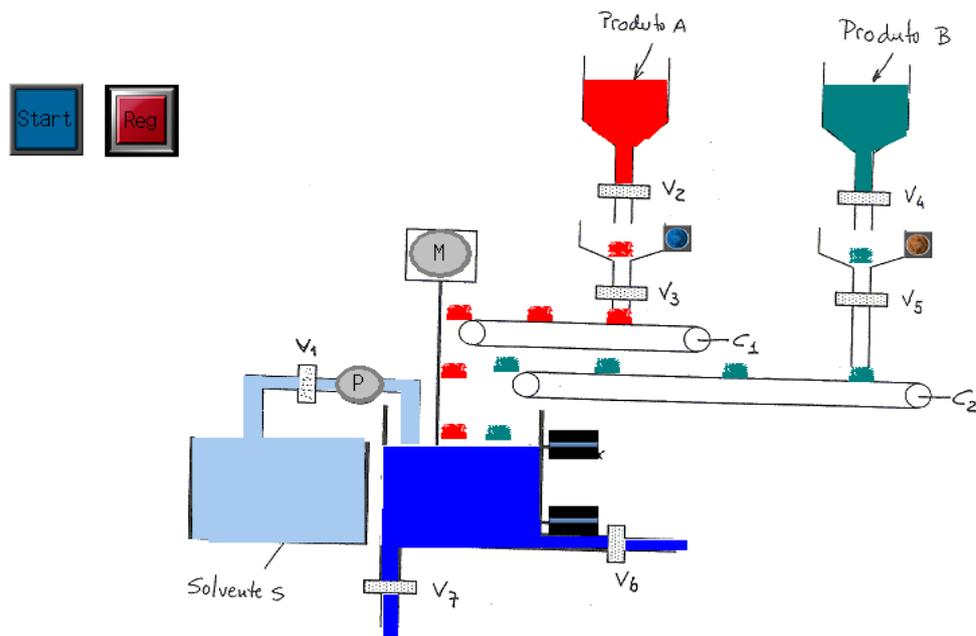


Figura 48 – Ambiente de simulação construído no CX-Designer.

Procedendo à simulação, o ambiente virtual vai-se comportar como ordenado pelo programa do controlador que tiver sido escrito no mesmo ficheiro do CX-Programmer utilizado para escrever as equações da parte operativa. Na figura 49 encontra-se representada uma imagem do ambiente de simulação virtual CX\_Designer recolhida ao longo da simulação.

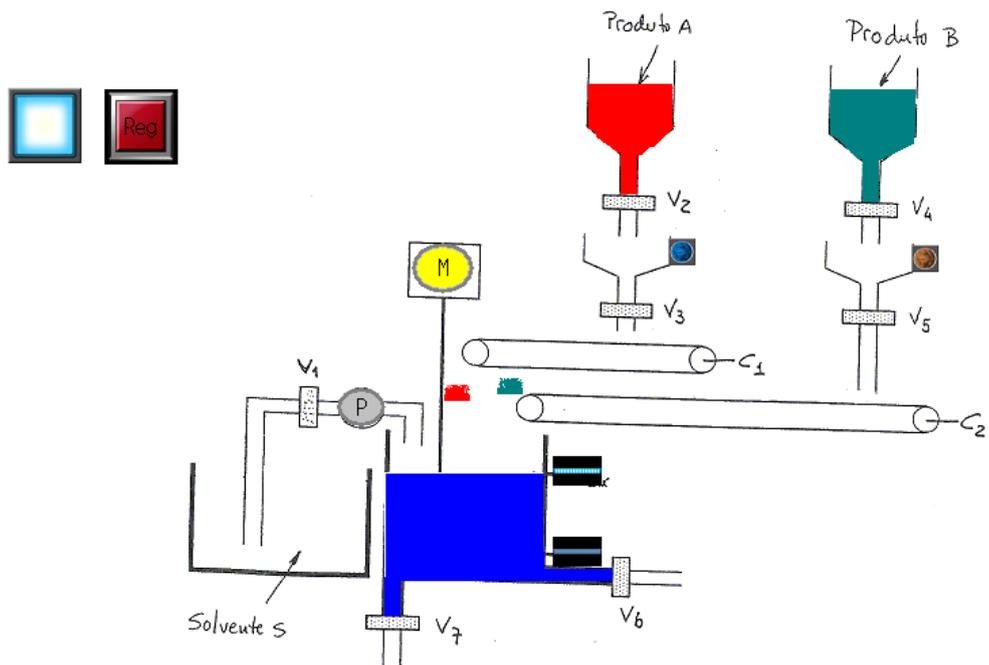


Figura 49 – Ambiente do CX-Designer ao longo da simulação.

### 5.3. Caso de estudo nº3 – Estação de transporte de objetos

Na Figura 50 encontra-se representado o sistema automatizado e modelar. Os carros C1 e C2 transportam peças entre os postos A1 e A2 e o posto B, partilhando o mesmo sistema de descarga comum em B. O movimento dos carros é o seguinte: com os dois carros nos postos de repouso (A1 e A2), após a ordem de partida (START), é dada a ordem AV2 ao carro C2, deslocando-se até B onde é descarregado. Terminada a operação de descarga, é dada a ordem REC2 para o carro C2 regressar ao posto A2. De seguida, é dada a ordem AV1 ao carro C1 para se deslocar até ao posto B onde é por sua vez, é descarregado. Após a ordem REC1, regressa ao posto A1. O sistema de descarga é constituído por uma pinça que pode subir (SP) e descer (DP), existindo dois sensores nas posições superior (se5) e inferior (se6). Pode ainda rodar à direita (RD) e rodar à esquerda (RE), podendo ser detetada a sua posição em cada 90° através dos sensores de posicionamento se1, se2, se3 e se4. O fecho da pinça (FP) é comandado por um êmbolo de efeito simples monoestável, sendo detetado pelo contacto pf. O ciclo de operações envolvido na descarga é o seguinte:

- Inicialmente a pinça está na posição superior direita e aberta;
- Após ser solicitada a descarga de um carro, a pinça desce, agarra o objeto, sobe, roda à esquerda e abre depositando o objeto no tapete de evacuação.

De seguida, regressa à posição inicial. Para ativar o tapete de evacuação é dada a ordem M. Cada carro possui um sensor que tem como função detetar quando se encontra um objeto no seu interior (fc1 para o C1 e fc2 para C2).

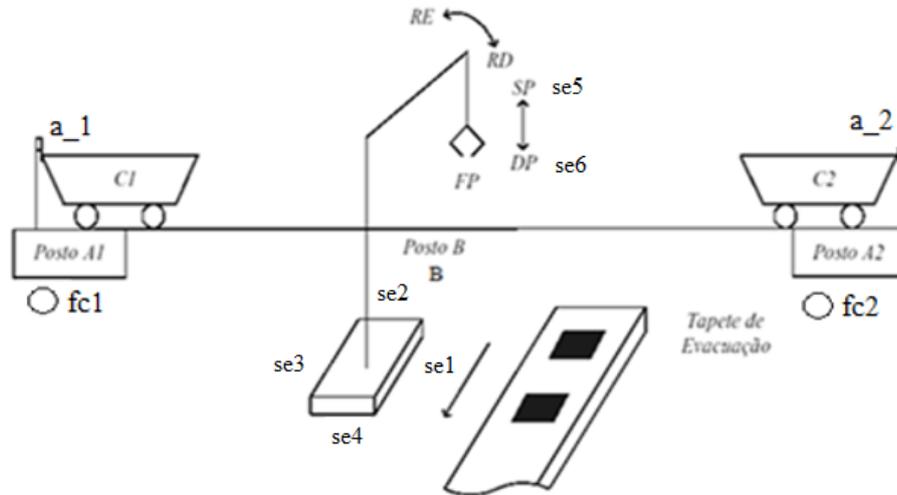


Figura 50 – Sistema automatizado a modelar.

### 5.3.1. Modelação do sistema automatizado

O sistema apresentado anteriormente possui um grande número e variedade de componentes físicos. Analisando o enunciado e a Figura 50 retiram-se algumas conclusões importantes, necessárias para a construção dos modelos:

1. Os carros C1 e C2 estão inseridos numa única pista, podendo movimentar-se em duas direções (direita e esquerda). Na realidade, se for dada a ordem pelo controlador para se movimentarem na direção um do outro, e ela persistir ao longo de um intervalo de tempo, eles em qualquer altura irão colidir num ponto da pista. Por esta razão, os modelos dos carros terão de ser restringidos, para que mesmo mantendo a ordem de avanço, não possam ocupar as mesmas posições simultaneamente na pista nem possam atravessar para as posições posteriores a estas;
2. Os sensores de posicionamento a\_1 e a\_2 só podem ser ativados pelos carros C1 e C2, respetivamente (justificado no ponto anterior). Porém, o sensor b irá ser ativado pelos dois carros quando cada um deles se encontrar no posto B. Devido a este motivo, este sensor deverá ser modelado duas vezes, tendo de pertencer aos modelos dos carros C1 e C2;

3. A garra terá de ser dividida em três componentes independentes, visto no seu conjunto poder realizar três movimentos distintos. Os componentes que a constituem são os seguintes:
  - Braço – Poderá rodar 360° para a direita ou esquerda. Os sensores s1, s2, s3 e s4 ficaram atuados dependendo da posição em que ele se encontra;
  - Guindaste – Poderá mover-se para cima e para baixo. Os sensores s5 e s6 serão dependentes deste componente, pois irão detetar as suas posições extremas (superior e inferior);
  - Pinça – a pinça será um atuador monoestável atuado pela ordem FP. O sensor pf irá detetar quando esta se encontra fechada, sendo por este motivo dependente da posição da pinça.
4. Os modelos observadores dos objetos são complexos de construir e têm de ser pensados ao pormenor. É necessário ter em atenção aos seguintes aspetos:
  - Os objetos a ser transportados do posto B para o tapete rolante pela garra têm origem em dois postos distintos, A1 e A2;
  - Enquanto a garra transporta um objeto para o tapete rolante, o carro que foi descarregado deve estar disponível para ser novamente carregado, e assim que contiver novamente um outro objeto, estar pronto a voltar para o posto B;
  - Quando o tapete se encontra a escoar as peças, deve ser possível à garra transportar mais peças, simultaneamente.

Atendendo aos aspetos apresentados nos pontos anteriores, foram elaborados os modelos para simular o comportamento do sistema automatizado.

#### 5.3.1.1. Tabelas de entradas e saídas do controlador

Entradas	Descrição	Word & Bit
START	Botão de acionamento do sistema	0.00
STOP	Botão de paragem do sistema	0.01
a_1	Carro no extremo direito da pista	0.02
a_2	Carro no extremo esquerdo na pista	0.03

b	Carro no meio da pista	0.04
pf	Pinça fechada	0.05
fc1	Objeto carregado no carro 1	0.06
fc2	Objeto carregado no carro 2	0.07
se1	Braço voltado para a direita	0.08
se2	Braço voltado para a frente	0.09
se3	Braço voltado para a esquerda	0.10
se4	Braço voltado para trás	0.11
se5	Pinça na posição extrema superior	1.00
se6	Pinça na posição extrema inferior	1.01

Tabela 14 – Entradas do controlador/saídas da parte operativa.

Saídas	Descrição	Word & Bit
SP	Subida da pinça	100.00
DP	Descida da pinça	100.01
FP	Fecho da pinça	100.02
RE	Rotação do braço para a esquerda	100.03
RD	Rotação do braço para a direita	100.04
AV1	Avanço do carro 1	100.05
REC1	Recuo do carro 1	100.06
AV2	Avanço do carro 2	100.07
REC2	Recuo do carro 2	101.00
M	Movimento do tapete para a frente	101.01

Tabela 15 – Saídas do controlador/entradas da parte operativa.

### 5.3.1.2. Modelos dos carros C1 e C2

Como ponto de partida para a modelação do sistema automatizado, foram elaborados os modelos dos carros C1 e C2, representados nas Figuras 51 e 52, respetivamente.

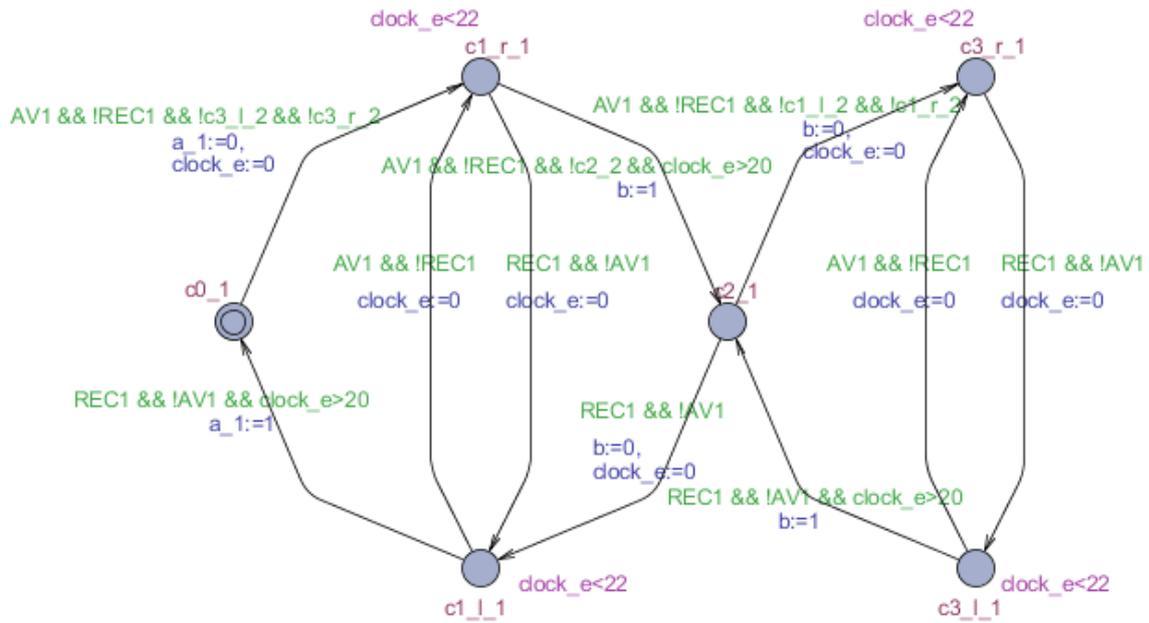


Figura 51 - Modelo construído para o carro C1.

Estados	Descrição
c0_1	Carro C1 na extremidade esquerda da pista (estado inicial)
c1_r_1	Carro C1 a deslocar-se para a direita vindo da posição inicial
c1_l_1	Carro C1 a deslocar-se para a esquerda vindo da posição intermédia
c2_1	Carro C1 a meio da pista (posição de interface com a garra)
c3_r_1	Carro C1 a deslocar-se para a direita vindo da posição intermédia
c3_l_1	Carro C1 a deslocar-se para a esquerda para a posição intermédia

Tabela 16 – Descrição de cada um dos estados do modelo do carro C1.

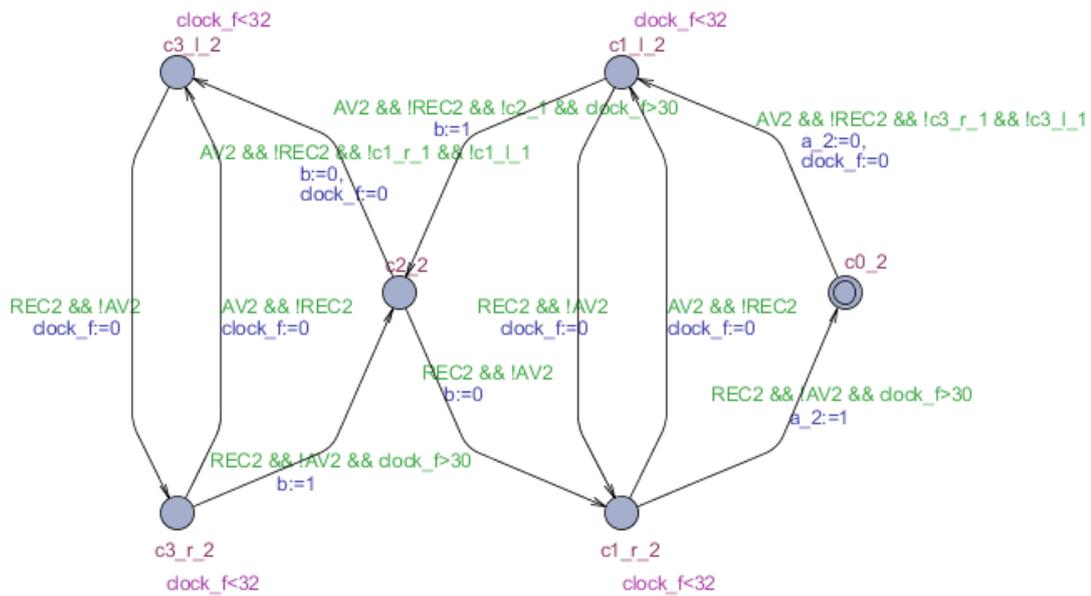


Figura 52 – Modelo construído para o carro C2.

Estados	Descrição
c0_2	Carro C2 na extremidade direita da pista (estado inicial)
c1_r_2	Carro C2 a deslocar-se para a direita vindo da posição intermédia
c1_1_2	Carro C2 a deslocar-se para a esquerda vindo da posição inicial
c2_2	Carro C2 a meio da pista (posição de interface com a garra)
c3_r_2	Carro C2 a deslocar-se para a direita para a posição intermédia
c3_1_2	Carro C2 a deslocar-se para a esquerda vindo da posição intermédia

Tabela 17 – Descrição de cada um dos estados do modelo do carro C2.

Como pode ser visto nas Figuras 51 e 52, cada um dos modelos dos carros possui dois estados intermédios entre cada estado onde se encontram os sensores de posicionamento. Apesar de estes representarem efetivamente a mesma posição na pista, possuem a função de distinguir o sentido de movimentação dos carros.

Além disto, os modelos foram restringidos para que estes não possam ocupar as mesmas posições na pista, simultaneamente. Por exemplo, observando a transição que liga o estado “c0\_1” ao estado “c1\_1\_1” no modelo do carro C1, é possível verificar que esta só será verdadeira quando os estados “c3\_1\_2” e “c3\_r\_2” não se encontram ativos

no modelo do carro C2. Estes modelos são também responsáveis pela atuação dos sensores de posicionamento localizados na pista (a\_1, a\_2 e b), através de atribuições de variáveis definidas nas transições entre os estados.

### **5.3.1.3. Modelos da garra**

Como foi referido anteriormente, para ser mais fácil modelar a garra do sistema automatizado, esta foi dividida nos três componentes que a constituem – braço, guindaste e pinça. Seguindo esta divisão, foram criados modelos distintos para cada um deles, para que no seu conjunto tenham a capacidade de simular corretamente o comportamento da garra, depois de receber as ordens do programa do controlador.

O primeiro modelo elaborado foi o do braço da garra e encontra-se representado na Figura 53. Tal como nos modelos dos carros C1 e C2, este modelo possui no total oito estados intermédios entre os estados onde se localizam os sensores, para que seja possível distinguir o sentido de rotação em que o braço se desloca. Além disto, este modelo tem também a função de ativar os sensores se1, se2, se3 e se4 (sensores de posicionamento do braço), através das atribuições de variáveis presentes nas transições entre os estados.

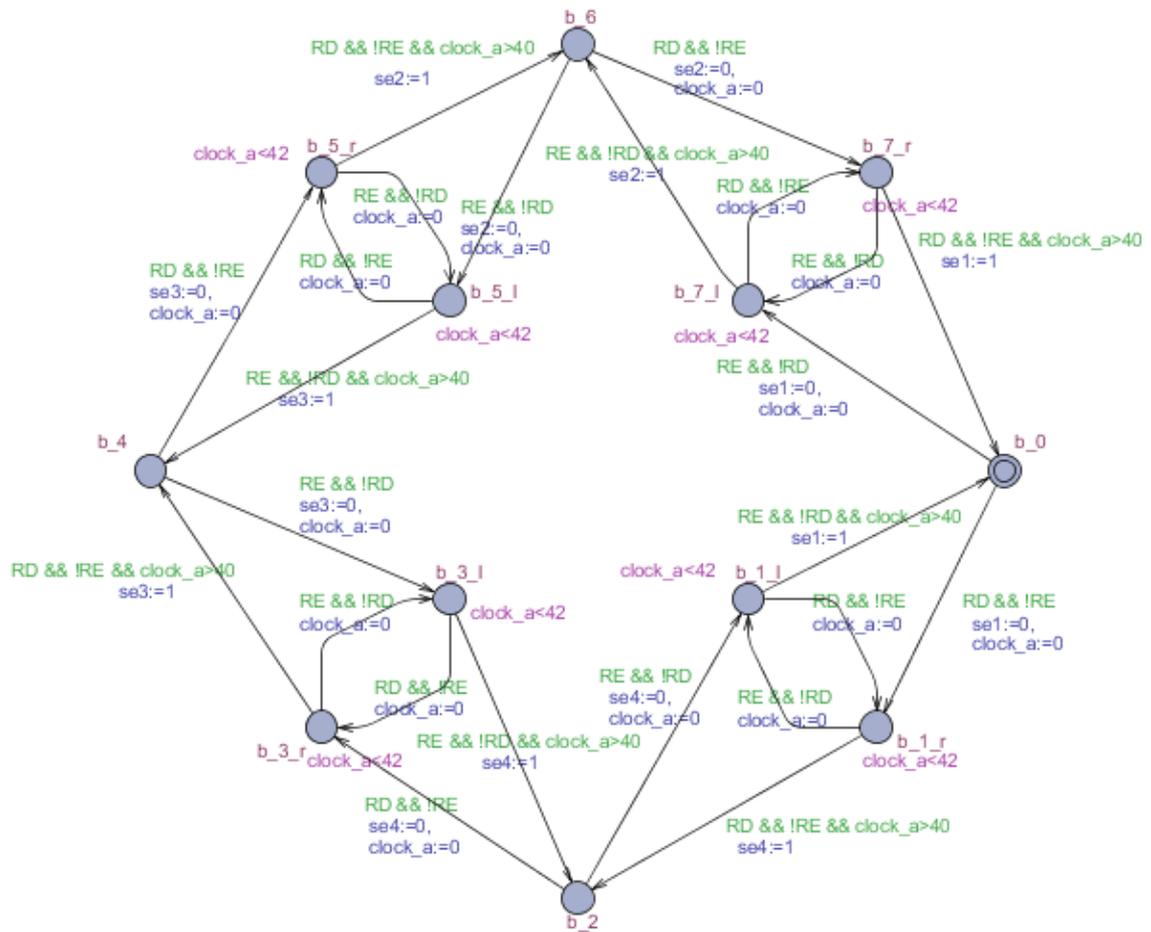


Figura 53 – Modelo do braço da garra.

Estados	Descrição
b_0	Garra voltada para a direita (estado inicial)
b_1_r	Garra a rodar para a direita (sentido horário)
b_1_l	Garra a rodar para a esquerda (sentido anti-horário)
b_2	Garra voltada para trás
b_3_r	Garra a rodar para a direita (sentido horário)
b_3_l	Garra a rodar para a esquerda (sentido anti-horário)
b_4	Garra voltada para a esquerda
b_5_r	Garra a rodar para a direita (sentido horário)
b_5_l	Garra a rodar para a esquerda (sentido anti-horário)
b_6	Garra voltada para à frente
b_7_r	Garra a rodar para a direita (sentido horário)
b_7_l	Garra a rodar para a esquerda (sentido anti-horário)

Tabela 18 – Descrição de cada um dos estados do modelo do braço da garra.

Para modelar o guindaste da garra foi criado o modelo de quatro estados representado na Figura 54. Este modelo permite ao guindaste mover-se para cima e para baixo, seja qual a posição em que o braço se encontre. Através das atribuições de variáveis, este modelo possui a função de ativar os sensores se5 e se6 nas posições superior e inferior, respetivamente.

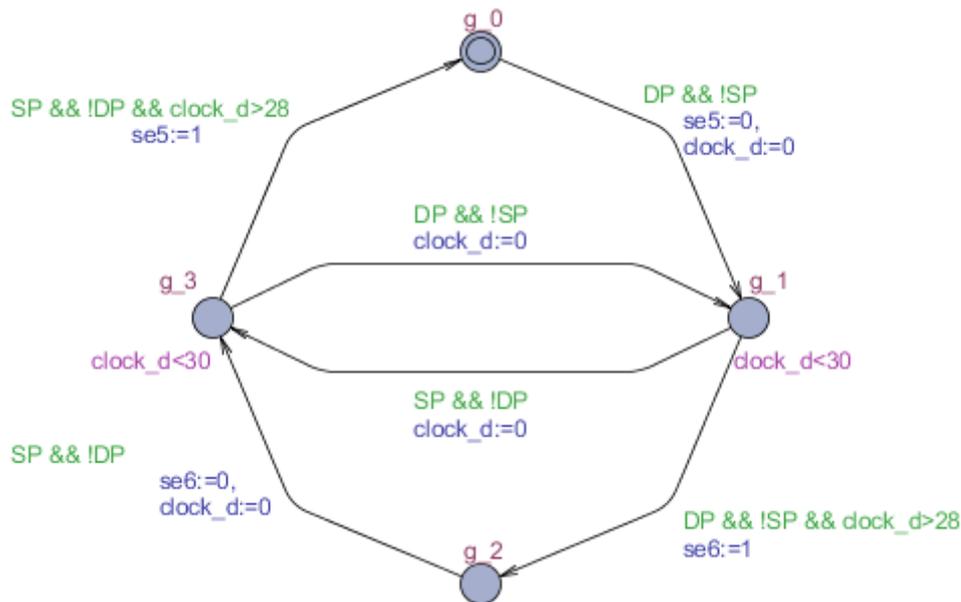


Figura 54 – Modelo do guindaste da garra.

Estados	Descrição
g_0	Garra na posição superior (estado inicial)
g_1	Garra a descer
g_2	Garra na posição inferior
g_3	Garra a subir

Tabela 19 - Descrição de cada um dos estados do modelo do guindaste da garra.

Finalmente, para completar a modelação da garra, foi criado o modelo da pinça representado na Figura 55. O modelo é de quatro estados (semelhante ao modelo do guindaste) e tem a função de ativar o sensor pf.

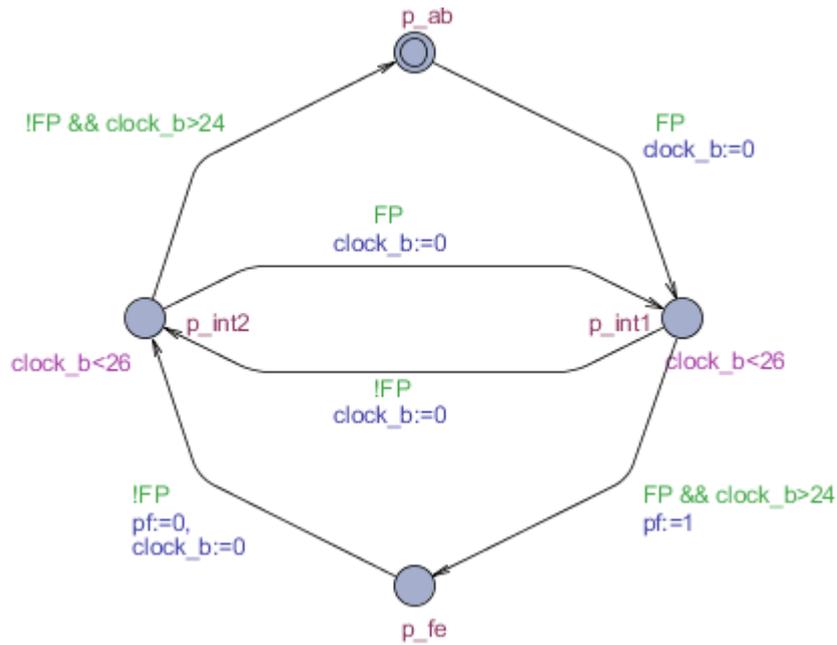


Figura 55 – Modelo da pinça.

Estados	Descrição
p_ab	Garra aberta (estado inicial)
p_int1	Garra a fechar
p_fe	Garra fechada
p_int2	Garra a abrir

Tabela 20 – Descrição de cada um dos estados do modelo da pinça da garra.

#### 5.3.1.4. Modelos dos objetos

Para modelar o comportamento dos objetos foram criados os modelos observadores representados na Figura 56. No total são quatro e em conjunto têm a função de representar o comportamento esperado que as peças tenham no sistema automatizado.

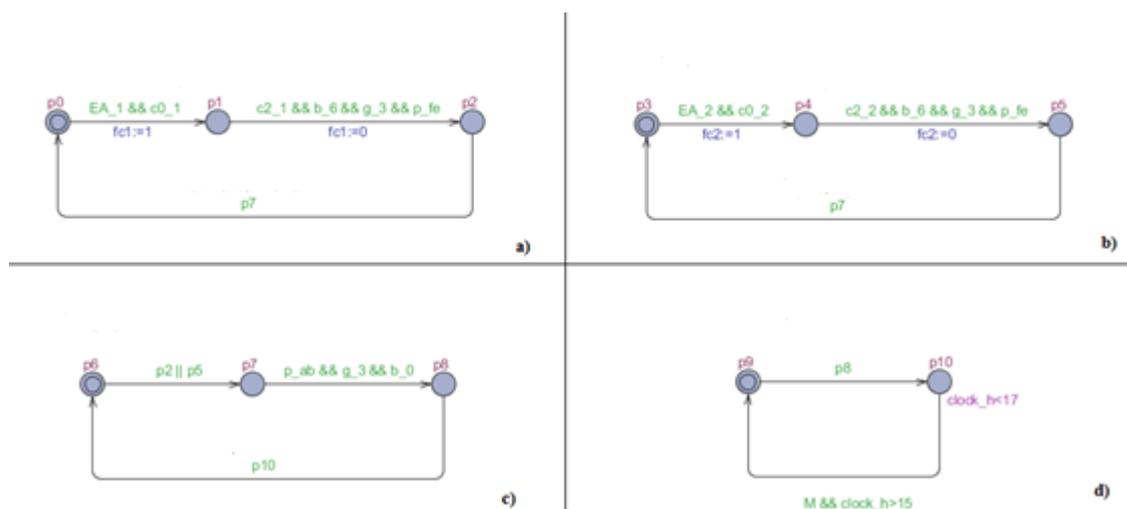


Figura 56 – Modelos observadores da peça: a) no carro C1, b) no carro C2, c) na grua, d) no tapete rolante.

Estados	Descrição
p0	Peça não carregada (estado inicial)
p1	Peça carregada
p2	Peça a sair do carro
p3	Peça não carregada (estado inicial)
p4	Peça carregada
p5	Peça a sair do carro
p6	Garra sem peça (estado inicial)
p7	Garra com peça (sentido horário)
p8	Peça a sair da garra
p9	Garra sem peça (estado inicial)
p10	Garra com peça (sentido horário)

Tabela 21 – Descrição de cada um dos estados dos modelos observadores da peça.

Foram criados quatro modelos distintos para ser possível obter uma simulação o mais próxima possível com o que acontece na realidade. Com esta divisão é possível existirem várias peças no sistema, simultaneamente. Assim, os carros podem ser carregados após terem sido descarregados pela garra ou, enquanto um objeto se encontra a ser evacuado pelo tapete rolante, a garra pode ir carregar novamente um outro logo que se encontre disponível nos carros.

Os modelos representados em a) e b) da Figura 56 tem a função de observar o comportamento dos objetos quando se encontram localizados nos carros C1 e C2,

respetivamente. Os estados “p0” e “p3” (estados iniciais de cada modelo) correspondem aos estados em que não existe nenhum objeto no interior dos carros. Eles são “carregados” através de eventos aleatórios distintos (*EA\_1* e *EA\_2*), que simulam o carregamento por parte de um operador, por exemplo.

O modelo representado em c) observa o objeto quando este se encontra agarrado pela garra. Como pode ser visto na figura, a garra tem a capacidade de agarrar objetos dos dois carros sem uma ordem pré-estabelecida. Além disto, enquanto não descarregar o objeto que se encontrar na sua posse (estado “p7” ativo), não pode agarrar outro.

Por fim, o modelo em d) representa o comportamento do objeto quando se encontra no tapete rolante. É um modelo de apenas de dois estados, com o objetivo de simular quando o objeto deixa de estar na garra para passar a estar no tapete no rolante e, ao fim de um tempo com a ordem M ligada, ser evacuado.

## **5.3.2. Simulação**

### **5.3.2.1. Alterações efetuadas nos modelos**

Antes de se proceder à simulação no CX-One, tal como nos outros casos de estudo apresentados anteriormente, foram realizadas alterações nos modelos apresentados.

O modelo do guindaste apresentado na Figura 54 simula corretamente o comportamento deste componente. No entanto, se fosse utilizado da forma como foi elaborado, vários “guindastes” iriam ser ligados aos mesmos estados, ativando simultaneamente quando um estado estivesse ativo. Isto iria gerar uma tremenda confusão no utilizador, ao observar a simulação a correr no CX\_Designer.

Para resolver este problema implementaram-se em conjunto as seguintes soluções:

- Restringir o modelo para que o guindaste só se possa movimentar nas posições do braço em que exerce trabalho (carga e descarga de objetos) conforme se encontra referido no enunciado do problema. Analisando o modelo do braço representado na Figura 53, estas posições correspondem aos estados “b\_0” e “b\_6”;
- Para cada posição do braço referida anteriormente, foi criado um modelo do guindaste.

No total foram criados dois novos modelos do guindaste, representados na Figura 57. Eles são em tudo semelhantes ao modelo apresentado na Figura 54, com a exceção de cada um deles possuir uma restrição que os impede de avançar enquanto o braço não se encontrar nas posições especificadas para a carga ou descarga dos objetos.

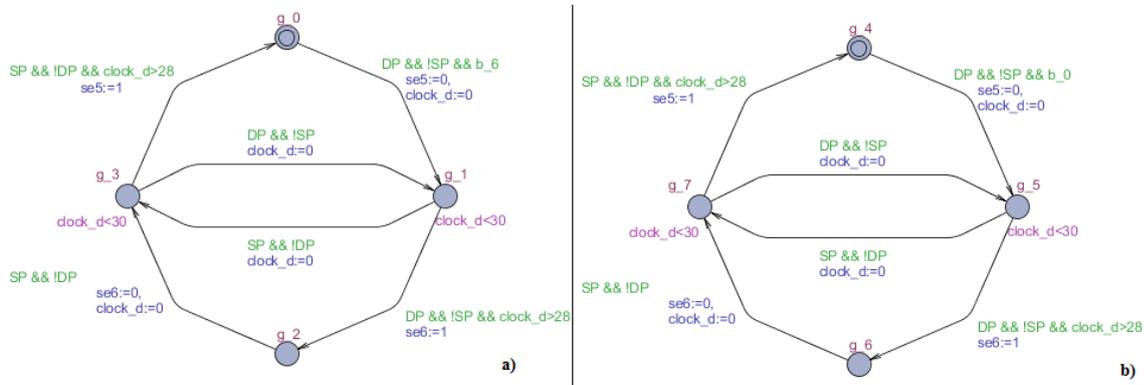


Figura 57 – Modelos do guindaste provenientes das alterações efetuadas para a simulação: a) modelo da carga, b) modelo da descarga.

Como consequência desta alteração realizada, outra alteração teve de ser feita num dos modelos observadores dos objetos, nomeadamente no modelo correspondente ao da garra. Visto os estados do modelo do guindaste elaborado para a operação de “carga” continuarem com a mesma nomenclatura, não foi necessário proceder alterações aos modelos observadores construídos para os carros C1 e C2. No entanto, o modelo do guindaste considerado para a “descarga” possui estados com designações diferentes, sendo então necessário atualizar a transição do modelo observador da garra que permite com que o objeto possa ser colocado no tapete rolante. O modelo em questão encontra-se representado na Figura 58, já com a transição devidamente corrigida.

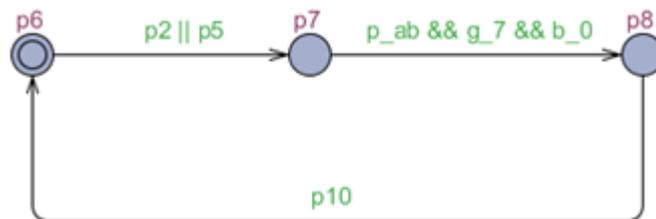


Figura 58 – Modelo observador da peça na garra corrigido.

### 5.3.2.2. Ambiente de simulação

Os modelos apresentados anteriormente foram traduzidos para equações da linguagem *ladder* e transcritos para o CX-Programmer, como se encontra representado na Figura 59. Por uma questão de espaço, é apenas apresentada uma figura de uma parte das equações obtidas.



Figura 59 – Representação parcial das equações ladder dos modelos construídos.

No processo de simulação, as equações representadas na figura anterior estarão ligadas pelo CX-Simulator ao ambiente virtual construído no CX-Designer, representado na Figura 60. Como é possível observar, este é um ambiente de simulação mais complexo, permitindo ao utilizador um desafio com maior grau de dificuldade.

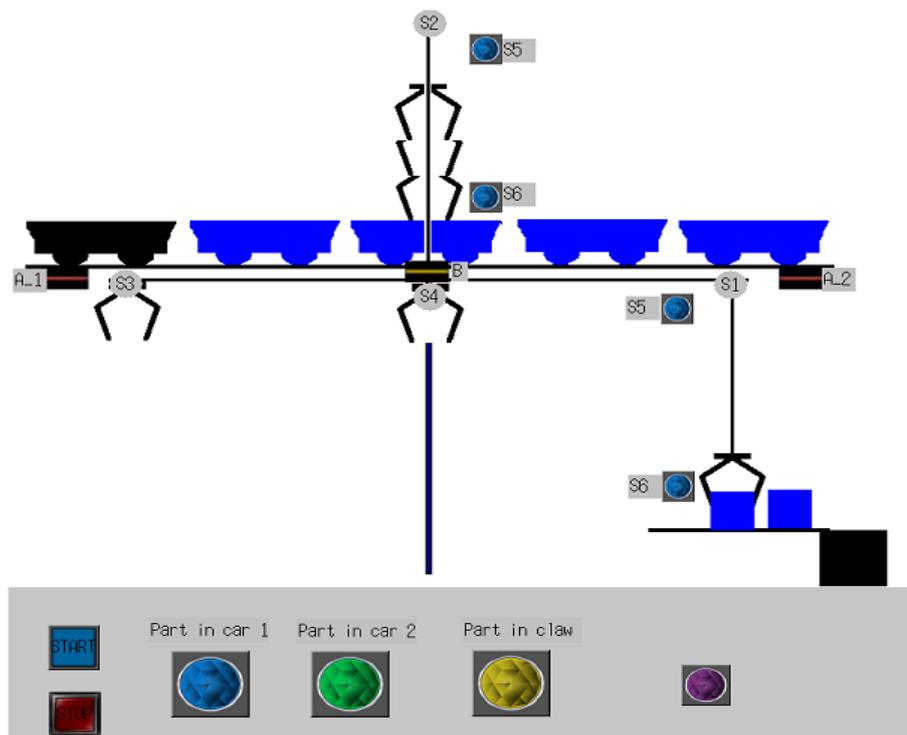


Figura 60 – Ambiente de simulação criado no CX-Designer.

Procedendo à simulação, o ambiente virtual vai-se comportar como ordenado pelo programa do controlador que tiver sido escrito no ficheiro do CX-Programmer, acima da parte operativa. Na Figura 61 encontra-se representada uma imagem do ambiente de simulação virtual CX\_Designer recolhida ao longo da simulação.

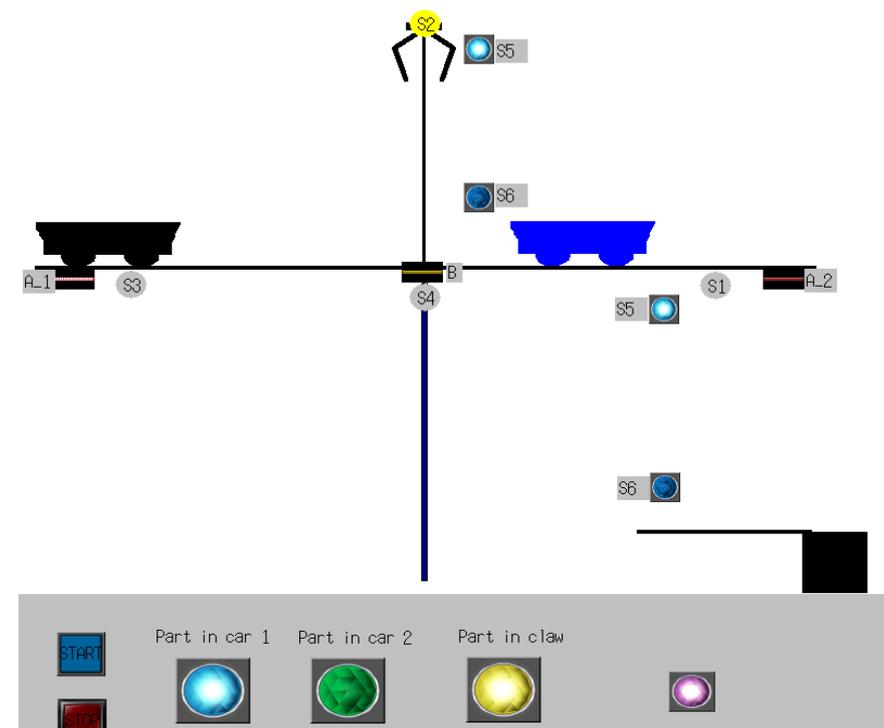


Figura 61 - Ambiente do CX-Designer ao longo da simulação.

## 5.4. Aplicação das simulações ao ensino da automação

A partir de uma definição das características pretendidas para uma bancada a desenvolver, o utilizador seguindo as regras anteriormente expostas pode construir e desenvolver os modelos da parte física do sistema e testar os seus programas de controlador.

Para o correto manuseamento destas ferramentas, deve existir algum cuidado na transcrição das equações *ladder* dos seus programas para o ambiente do CX-Programmer. Devem ser utilizados os mesmos endereços do PLC virtual para as entradas e saídas dos programas que se encontram referidos nas “tabelas de entradas e saídas do controlador”, anexadas a cada uma das bancadas de simulação. Só desta forma é que o programa do controlador poderá estabelecer a ligação aos modelos da parte operativa, necessária para a sua interação. Além disto, é imperativo que para os restantes aspetos dos programas do controlador (condições de inicialização, etapas, condições de transposição de etapa, etc.) sejam utilizados os endereços entre as *words* 2.00 e 99.15 do PLC virtual, para que não exista a possibilidade de existir conflito com os endereços utilizados para escrever as equações da parte operativa.

Para proceder à simulação, os ficheiros do CX-Programmer e do CX-Designer devem estar abertos no computador, simultaneamente. A simulação deve ser posta a correr a partir da barra de ferramentas do CX-Designer.

## **5.5 Informação a reter do capítulo**

Neste capítulo foram apresentados três casos de estudo distintos, onde foram explicadas pormenorizadamente todas as tarefas realizadas para a construção das respetivas bancadas de simulação, a partir do enunciado de um exercício.

Para a sua utilização, o utilizador deve construir programas de controlador, convertê-los para linguagem *ladder* e escrevê-los no ficheiro do CX-Programmer, por cima das equações da parte operativa. Só depois deste trabalho realizado, é que estão em condições de visualizar a simulação a partir do ambiente do CX-Designer. Nos casos de estudo apresentados, a modelação dos respetivos controladores foi feita por um utilizador diferente [50], para garantir maior robustez da solução encontrada.



# **Capítulo 6**

# **CONCLUSÃO E**

# **TRABALHOS FUTUROS**



## 6. CONCLUSÃO E TRABALHOS FUTUROS

Para esta dissertação foram propostos três objetivos distintos, com diferentes graus de importância. O objetivo principal consistiu na construção do máximo número de bancadas de simulação utilizando a abordagem de simulação MiL. Os outros dois objetivos, mais específicos, incidiram na elaboração de ferramentas direcionadas para trabalhos futuros nesta área. O primeiro consistiu na elaboração de uma metodologia eficaz que permitisse a qualquer utilizador, com mínimos conhecimentos na área da modelação, construir uma bancada de simulação a partir de um enunciado e/ou esquema de um problema proposto. Por sua vez, o último objetivo consistiu na construção de uma base de dados de modelos de componentes físicos, aproveitando desta forma todo o trabalho para o objetivo principal.

Para a modelação dos sistemas automatizados foram explorados vários formalismos, tendo sido escolhido os autómatos finitos temporizados. Esta escolha deveu-se principalmente ao seu conhecimento adquirido ao longo ciclo de estudos do mestrado, da informação disponibilizada pelo orientador e ao fato de ser uma ferramenta onde não é necessário proceder a alterações dos modelos para a realização da verificação formal, num *software* específico para o efeito.

Como *software* de construção e simulação das bancadas de simulação foi utilizado o CX-One da OMRON. Este foi escolhido devido à sua utilização em trabalhos na área realizados anteriormente, tendo por isso já apresentado garantias de um bom funcionamento.

O objetivo principal foi atingido com sucesso. No total foram construídas catorze bancadas de simulação, tendo sido devidamente simuladas. Nos casos de estudo desta dissertação são apresentadas três das bancadas de simulação construídas através de exercícios propostos, onde estão definidos os constituintes dos respetivos sistemas automatizados. Na parte da simulação, a parte de comando é modelada em função de um determinado problema, só a título exemplificativo. Já a parte operativa, é modelada em função dos elementos que o sistema possui, podendo esta interagir com diferentes programas do comando, equiparando-se assim a um sistema automatizado real.

Relativamente aos objetivos específicos propostos, foram também alcançados. Foi desenvolvida e proposta uma metodologia que pode ser facilmente seguida para a elaboração de uma bancada de simulação. Além disso, todos os modelos construídos

foram guardados e agrupados num manual anexado a esta dissertação, podendo no futuro serem verificados e reutilizados para os fins mais diversos.

Para trabalhos futuros, é sugerido que seja feita uma verificação formal de todos os modelos das bancadas de simulação construídas. Além disto, a metodologia apresentada nesta dissertação deve ser utilizada para o desenvolvimento de novas bancadas, com desafios maiores e próximos do que existe atualmente na indústria.

## REFERÊNCIAS

- [1] E. Matias, P. Oliveira, J. Cunha, E. Pires e F. Soares, “E-GRAF CET: A MULTIMEDIA EDUCATIONAL TEACHING TOOL,” *Controlo 2010, 9th Portuguese Conference on Automatic Control*, Coimbra, Portugal, 8-10 Setembro, 2010.
- [2] E. Hansen, “The role of interactive video technology in higher education: Case study and proposed framework”, *Education Technology*, September 1990, 13-21.
- [3] N. Carvalho, R. Silveira, C. Leão, J. Machado and F. Soares, “Platform WALC: design and development of a PLC network,” December 10-11, 2009, Bratislava, Slovak Republic, E-Academia Slovaca, N.O. ISBN: 978-80-89316-11-3.
- [4] F. Mateos, “Estandarización con autómatas programables – Sistema automatizado PLCs”, *Universidad de Oviedo, Espana*, 2001.
- [5] J. Ferreira, “Modelação de Sistemas Hidráulicos para Simulação com Hardware-in-the-loop”, *Tese de Doutoramento, Universidade de Aveiro, Portugal*, 2003.
- [6] Apresentação na internet: “Physical System Modeling”. Endereço web: [www.iaa.ncku.edu.tw/~clchen/course/control/Modelling.pdf](http://www.iaa.ncku.edu.tw/~clchen/course/control/Modelling.pdf), consultada em Setembro de 2013.
- [7] Página da internet: Modelica. Endereço web: [www.modelica.org](http://www.modelica.org), consultada em Setembro de 2013.
- [8] Página da internet: Dymola. Endereço web: [www.dymola.com](http://www.dymola.com), consultada em Setembro de 2013.
- [9] Página da internet: Modelica Association. Endereço web: [www.modelica.org/association](http://www.modelica.org/association), consultada em Setembro de 2013.
- [10] Página da internet: OpenModelica. Endereço web: <https://www.openmodelica.org/>, consultada em Setembro de 2013.
- [11] H. Elmqvist, S. Mattsson, “An introduction to the physical modeling language modelica”, *Proceedings of the 9th European Simulation Symposium, ESS'97, Oct 19-23, 1997, Passau, Germany*.
- [12] J. Ramos, “Introdução à modelação de sistemas utilizando redes de Petri”, *Instituto Politécnico de Beja, Escola Superior de Tecnologia e Gestão*, 6 de Janeiro de 2001.
- [13] M. Cunha, “Redes de Petri”, *Departamento de Informática, Universidade do Minho*, 2005.
- [14] N. Bertrand, “Timed Automata”, *VerTeCs, INRIA Rennes*, 2009.
- [15] N. Canadas, C. Barros, J. Machado, F. Soares, “A Simulation Platform for Automation Teaching”, *CISPEE2013, Porto*, 2013.
- [16] J. Machado, “Influence de la prise en compte d’un modele de processus en vérification formelle des Systèmes à Evénements Discrets”, *Tese de Doutoramento em Engenharia Mecânica, Ecole Normale Supérieure de Cachan*, France, Maio 2006.

- [17] C. Pegden, R. Sadowski, "Introduction to Simulation Using Siman", Second Edition, McGraw-Hill, January 1991.
- [18] J. Machado, E. Seabra, J. Campos, C. Leão e F. Soares, "Safe controllers design for industrial automation systems", Computer and industrial Engineering journal, Elsevier, 2010.
- [19] D. Chioran, J. Machado, "Design of a Mechatronic System for Application of Hardware-in-the-loop Simulation Technique", MECAHITECH'11 International Conference, Bucharest, 22-23 September 2011.
- [20] S. Demers, P. Gopalakrishnan, L. Kant, "A Generic Solution to Software-in-the-loop", Applied Research, Telcordia Technologies, 331 Newman Springs Road, Red Bank, USA.
- [21] O. Meister, N. Frietsch, J. Seibold, G. Trommer, "Software-in-the-loop simulation for small autonomous VTOL UAV with teaming capability", Institute of Systems Optimization, University of Karlsruhe, Germany.
- [22] Página da internet: Mathworks HiL simulation. Endereço web: <http://www.mathworks.com/verification-validation/hil-simulation.html>, consultada em Setembro de 2013.
- [23] Jim A. Ledin, "Embedded Systems Programming: Hardware-in-the-loop simulation", Fevereiro 1999, 42 – 60.
- [24] D. Ramaswamy, R. McGee, S. Sivashankar, A. Deshpande, J. Allen, K. Rzemien, W. Stuart, "A case study in hardware-in-the-loop testing: development of an ECU for a hybrid electric vehicle", in "Software/Hardware Systems, Systems Engineering, Advanced Electronics Packaging, and Electromagnetic Compatibility" (SAE Special Publication Papers, 2004).
- [25] J. Keranen, T. Raty, "Model-based testing of embedded systems in hardware in the loop environment", VTT Technical Research Centre of Finland, Kaitova yla 1, P.O. Box 1100, Oulu 90571, Finland.
- [26] O. Ljungkrantz, K. Akesson, M. Fabian, Y. Chengyin (2010). "Formal Specification and Verification of Industrial Control Logic Components". IEEE Transactions on Automation Science and Engineering. 2010, 7; 3, 538 – 548.
- [27] Página da internet: LMS simulation. Endereço web: <http://www.lmsintl.com/mil-sil-hil-and-real-time>, consultada em Setembro de 2013.
- [28] W. Zhu, S. Pekarek, J. Jatskevish, O. Wasynczuk, D. Delisle, "A Model-in-the-Loop Interface to Emulate Source Dynamics in a Zonal DC Distribution System", IEEE TRANSACTIONS ON POWER ELECTRONICS, VOL. 20, NO. 2, MARCH 2005, 438 – 445.
- [29] P. Fritzson, "Principles of Object-Oriented Modeling and Simulation with Modelica" 2.1, 939 pages, Wiley-IEEE Press, ISBN 0-471-471631.
- [30] J. Dantas, "Software de Simulação Hardware-in-the-Loop para a Simulação do Sistema de Navegação e Controle de Veículos Autônomos Submarinos", Escola Politécnica da Universidade de São Paulo, 2008.

- [31] O. Özgün, Y. Barlas, “Discrete vs. Continuous Simulation: When Does It Matter?”, Proceedings of the 27th International Conference of The System Dynamics Society, July 26 – 30, 2009, Albuquerque, NM, USA.
- [32] Página da internet: Dymola. Endereço web: <http://www.modelon.com/products/dymola/>, consultada em Agosto de 2013.
- [33] Página da internet: Dymola. Endereço web: <http://epg.eng.ox.ac.uk/content/dymola>, consultada em Agosto de 2013.
- [34] M. Dempsey, “Dymola for Multi-Engineering Modelling and Simulation”, Claytex Services Limited, Unit 9A, Nunhold Business Centre, Dark Lane, Hatton, United Kingdom.
- [35] “What is Dymola”, Dymola tutorial.
- [36] Página da internet: Simulink. Endereço web: <http://www.mathworks.com/products/simulink/>, consultada em Setembro de 2013.
- [37] Página da internet: Mathworks. Endereço web: <http://www.mathworks.com/>, consultada em Setembro de 2013.
- [38] Página da internet: CX-One. Endereço web: [http://industrial.omron.eu/en/products/catalogue/automation\\_systems/software/programming/cx-one/default.html](http://industrial.omron.eu/en/products/catalogue/automation_systems/software/programming/cx-one/default.html), consultada em Agosto de 2013.
- [39] Página da internet: OMRON. Endereço web: <http://www.omron.com/>, consultada em Agosto de 2013.
- [40] Manual do CX-One, disponível em [www.omron.com](http://www.omron.com).
- [41] Página da internet: Catálogo do CX-One. Endereço web: [http://industrial.omron.pt/pt/products/catalogue/automation\\_systems/software/configuration/cx-one/default.html](http://industrial.omron.pt/pt/products/catalogue/automation_systems/software/configuration/cx-one/default.html), consultada em Dezembro de 2012.
- [42] K. John, M. Tiegelkamp, “IEC 61131-3: Programming Industrial Automation Systems”, Springer-Verlag, Berlin, Germany, 2001.
- [43] IEC - International Electrotechnical Commission. “Programmable Controllers - Part 3.” CEI 61 131-3, 1993.
- [44] “CX-Programmer User Manual”, disponível em [www-omrom.com](http://www-omrom.com).
- [45] Página da internet: CX-Programmer. Endereço web: [http://industrial.omron.pt/pt/products/catalogue/automation\\_systems/software/configuration/cx-one/cx-programmer.html](http://industrial.omron.pt/pt/products/catalogue/automation_systems/software/configuration/cx-one/cx-programmer.html), consultada em Dezembro de 2012.
- [46] Página da internet: CX-Simulator. Endereço web: [http://industrial.omron.pt/pt/products/catalogue/automation\\_systems/software/configuration/cx-one/cx-simulator.html](http://industrial.omron.pt/pt/products/catalogue/automation_systems/software/configuration/cx-one/cx-simulator.html), consultada em Dezembro de 2012.

[47] Página da internet: CX-Designer. Endereço web:

[http://industrial.omron.pt/pt/products/catalogue/automation\\_systems/software/programming/cx-one/cx-designer.html](http://industrial.omron.pt/pt/products/catalogue/automation_systems/software/programming/cx-one/cx-designer.html), consultado em Dezembro de 2012.

[48] “CX-Designer User Manual”, disponível em [www.omron.com](http://www.omron.com).

[49] M. Uzam, “A general technique for the PLC-Based implementation of RW supervisors with time delay functions”. *Int J Adv Manuf Technol* (2012) 62:687-704. DOI 10.1007/s00170-011-3817-1.

[50] C. Barros, “Desenvolvimento de plataformas de automação digitais”, Dissertação de Mestrado Integrado em Engenharia Mecânica, Universidade do Minho, Guimarães, Portugal, 2013.



