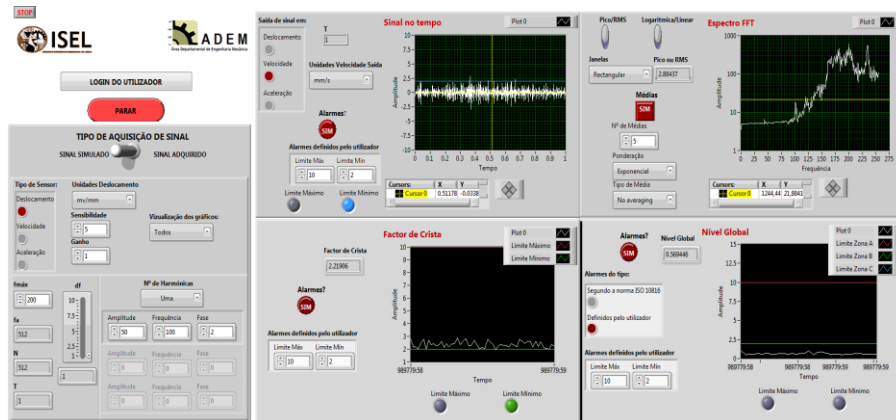




ISEL

INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA
Área Departamental de Engenharia Mecânica



Diagnóstico de Falhas Via Internet de Processos Industriais Controlados Com Autómatos Programáveis

ÍRIS MARGARIDA MOREIRA SANTOS NETO
(Licenciada em Engenharia Mecânica)

Trabalho Final de Mestrado para obtenção do grau de Mestre em Engenharia Mecânica

Orientador:

Doutor Mário José Gonçalves Cavaco Mendes

Júri:

Presidente: Doutor João Manuel Ferreira Calado

Vogais:

Doutor Luís Filipe Figueira Brito Palma

Doutor Mário José Gonçalves Cavaco Mendes

Novembro de 2016



ISEL

INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA
Área Departamental de Engenharia Mecânica

Diagnóstico de Falhas Via Internet de Processos Industriais Controlados Com Autómatos Programáveis

ÍRIS MARGARIDA MOREIRA SANTOS NETO
(Licenciada em Engenharia Mecânica)

Trabalho Final de Mestrado para obtenção do grau de Mestre
em Engenharia Mecânica

Orientador:

Doutor Mário José Gonçalves Cavaco Mendes

Júri:

Presidente: Doutor João Manuel Ferreira Calado

Vogais:

Doutor Luís Filipe Figueira Brito Palma

Doutor Mário José Gonçalves Cavaco Mendes

Novembro de 2016

Agradecimentos

Em primeiro lugar, quero agradecer ao Doutor Mário J. G. C. Mendes pela orientação, ajuda, apoio, disponibilidade e compreensão ao longo desta jornada.

Gostaria de agradecer ao Engenheiro Paulo Almeida pelo auxílio na resolução de alguns problemas de nível técnico que foram surgindo durante a execução deste Trabalho Final de Mestrado.

Um especial agradecimento ao meu pai que me acompanhou durante todo o meu percurso acadêmico, dando apoio emocional e criando as condições para que eu pudesse terminar o mesmo.

À minha mãe, que infelizmente já não se encontra presente fisicamente para testemunhar o encerrar deste capítulo, mas que muita força me transmitiu até aos últimos momentos da sua existência.

Resumo

Actualmente, devido à competitividade global e à necessidade de redução de custos, as empresas têm cada vez mais interesse em aumentar a disponibilidade dos seus sistemas de produção. Além disso, a Internet é cada vez mais indispensável para a distribuição de informação em todo o mundo, tornando-se decisivo implementar sistemas de supervisão à distância e que detectem e diagnostiquem falhas. Assim, surgem os conceitos de *Industrial Internet of Things* e de Indústria 4.0, em que a partir da grande quantidade de informação disponível rapidamente podem ser tomadas acções correctivas, impedindo a ocorrência de falhas e avarias.

Por esta razão, o sistema proposto neste trabalho é um sistema de detecção e diagnóstico de falhas via Internet, assente na detecção através do método de verificação de limites fixos e no diagnóstico a partir de árvores de falha com lógica booleana e regras IF-THEN, com a verificação das variáveis lógicas de entrada e saída. Este sistema tem aplicabilidade num vasto número de processos industriais e foi testado no laboratório remoto LabAPI – ISEL. Adicionalmente e dado que, a vibração mecânica ocorre com frequência em vários processos e equipamentos presentes na indústria, procedeu-se à implementação de um sistema de monitorização e análise de vibrações. O sistema de análise de vibrações inclui um espectro da transformada rápida de Fourier e alarmes relativos ao valor eficaz.

Este sistema foi elaborado com recurso a alguns programas, nomeadamente o programa de linguagem gráfica LabVIEW, utilizando o módulo LabVIEW Datalogging and Supervisory Control e o OPC Servers para comunicação entre os autómatos programáveis Siemens S7-1200 e o sistema de detecção e diagnóstico de falhas proposto.

Palavras-chave

Diagnóstico de Falhas; Internet; Autómato Programável.

Abstract

Currently, due to the global competitiveness and the need for cost reduction, companies are more and more interested in increasing the availability of their production systems. Moreover, Internet is becoming indispensable for the distribution of information throughout the world, becoming critical to implement distance supervision systems which detect and diagnose faults. Thus, the concepts of Industrial Internet of Things and Industry 4.0 arise, where from a large amount of available information corrective actions can be quickly taken, preventing the occurrence of faults and failures.

For this reason, the system proposed in this paper is an Internet based fault detection and diagnosis system established by the fixed threshold checking detection method and decision trees with boolean logic and IF-THEN rules diagnosis, to check input and output logical variables. This system has applicability in a wide range of industrial processes and has been tested on the remote laboratory LabAPI – ISEL. In addition and given that mechanical vibration occurs frequently in various processes and equipment present in industry, it was implemented a vibration monitoring and analysis system. The vibration analysis system includes a Fast Fourier Transform spectrum and alarms for the root mean square value.

This system was developed using a number of software, including the graphical programming software LabVIEW using the Datalogging and Supervisory Control LabVIEW module and the OPC Servers for the communication between Siemens S7-1200 programmable controllers and the monitoring and fault detection system.

Keywords

Fault Diagnosis; Internet; Programmable Controller.

Glossário

ARMA	Autoregressive-Moving Average (Média Móvel Auto-Regressiva)
CPS	Cyber-Physical Systems (Sistemas Ciber-Físicos)
DAQ	Data Acquisition (Aquisição de Dados)
ETA	Event-tree Analysis (Análise de Árvore de Eventos)
FDI	Fault Detection and Isolation (Detecção e Isolamento de Falhas)
FFT	Fast Fourier Transform (Transformada Rápida de Fourier)
FTA	Fault-tree Analysis (Análise de Árvore de Falhas)
HMI	Human-Machine Interface (Interface Humano-Máquina)
HTML	HyperText Markup Language (Linguagem de Marcação de Hipertexto)
IIoT	Industrial Internet of Things (Internet das Coisas na Indústria)
I/Os	Inputs and Outputs (Entradas e Saídas)
ISEL	Instituto Superior de Engenharia de Lisboa
NI	National Instruments
OLE	Object Linking Embedding
OPC	Open Platform Communications
PAC	Programmable Automation Controller (Controlador de Automação Programável)

PBC	Percentagem de Banda Constante
PLC	Programmable Logic Controller (Controlador Lógico Programável)
RMS	Root Mean Square (Valor Eficaz)
RTU	Remote Terminal Unit (Unidade Terminal Remota)
SCADA	Supervisory Control and Data Acquisition (Sistema de Supervisão e Aquisição de Dados)
STFT	Short-time Fourier Transform (Transformada de Fourier Por Janelas)
SVE	Shared Variable Engine (Motor da Variável Partilhada)
UA	Unified Architecture (Arquitetura Unificada)
VI	Virtual Instrument (Instrumento Virtual)
VPN	Virtual Private Network (Rede Privada Virtual)

Índice

Agradecimentos	i
Resumo	iii
Palavras-chave	iii
Abstract	v
Keywords	v
Glossário	vii
Índice	ix
Lista de Figuras	xiii
Lista de Tabelas	xvii
Lista de Símbolos	xix
1. Introdução	1
1.1. Motivação	1
1.2. Objectivos	4
1.3. O Sistema Proposto de Diagnóstico de Falhas Via Internet	4
1.4. Estrutura do Documento	5
2. Estado da Arte	7
2.1. Indústria 4.0	7
2.2. Monitorização, Detecção e Diagnóstico de Falhas Via Internet	11
2.2.1. Falhas, Avarias e Erros	12
2.2.2. Classificação de Falhas	13
2.3. Detecção e Diagnóstico de Falhas	16
2.3.1. Métodos de Detecção de Falhas	17
2.3.2. Métodos de Diagnóstico de Falhas	22
2.4. Monitorização e Análise de Vibrações	26
2.4.1. Conceitos de Vibração	27

2.4.2.	Aquisição e Processamento de Sinal	30
2.4.3.	Espectro FFT	33
2.4.4.	Outras Técnicas de Processamento de Sinal	39
2.5.	Trabalho Existente em Detecção e Diagnóstico de Falhas Via Internet	41
3.	Sistema de Detecção e Diagnóstico de Falhas via Internet Proposto	47
3.1.	Programas e Linguagens de Programação Utilizados	47
3.1.1.	National Instruments LabVIEW	48
3.1.2.	OPC Servers	49
3.2.	Aquisição de Dados – NI USB-6008	50
3.3.	Arquitectura do Sistema Proposto	54
3.3.1.	Arquitectura do Programa	58
4.	Resultados Experimentais	63
4.1.	Configuração das Etiquetas OPC	63
4.2.	Autenticação do Utilizador	65
4.3.	Experiência “Pórtico”	71
4.4.	Experiência “Cilindros”	80
4.4.1.	Pressão da Rede Pneumática	82
4.4.2.	Variáveis Lógicas	84
4.5.	Análise de Vibrações	89
4.5.1.	Configuração dos Dados de Aquisição	89
4.5.2.	Sinal no Tempo	94
4.5.3.	Espectro FFT	98
4.5.4.	Factor de Crista	104
4.5.5.	Nível Global	106
5.	Conclusões	111
5.1.	Trabalho Final	111
5.2.	Trabalho Futuro	112

Referências Bibliográficas	115
Anexo A	123
Anexo B	125

Lista de Figuras

Figura 2.1: Evolução da indústria até à Indústria 4.0; adaptado de [13].	8
Figura 2.2: Arquitectura do sistema SCADA do Departamento de Engenharia Mecatrónica da Faculdade de Engenharia de Kongu, Índia; adaptado de [18].	10
Figura 2.3: Tipos de falhas num sistema.	13
Figura 2.4: Tipo de falhas quanto à sua origem, adaptado de [26].	14
Figura 2.5: Falhas de natureza temporal – a) abruptas; b) incipientes; c) intermitentes; adaptado de [10].	15
Figura 2.6: Classificação de falhas utilizada em métodos de FDI baseados em modelos de processo – a) aditivas; b) multiplicativas; adaptado de [10].	15
Figura 2.7: Métodos de detecção de falhas; adaptado de [10].	18
Figura 2.8: Verificação de limites – a) limites fixos; b) limites de tendência; adaptado de [10].	19
Figura 2.9: Esquema geral do método de detecção de falhas baseado em modelos de processo; adaptado de [10].	20
Figura 2.10: Esquema geral do método de detecção de falhas baseado em modelos de sinal; adaptado de [10].	21
Figura 2.11: Métodos de diagnóstico de falhas adaptado de [10].	23
Figura 2.12: Métodos de inferência para o diagnóstico de falhas a) redes causais b) árvores de falha [34].	25
Figura 2.13: Três passos num sistema de detecção e diagnóstico de falhas.	27
Figura 2.14: Vibração harmónica.	28
Figura 2.15: Vibração periódica representada no domínio do tempo e na frequência do sinal.	30
Figura 2.16: Exemplo de sinal no tempo.	33
Figura 2.17: Sinal representado no domínio do tempo e no domínio da frequência; adaptado de [49].	34
Figura 2.18: Exemplo do efeito de janela num sinal não-periódico no intervalo de tempo [49].	36

Figura 2.19: Atenuação do fenómeno de <i>leakage</i> com a aplicação de janela apropriada (adaptado de [49]).	37
Figura 2.20: Aplicação da janela Rectangular em regimes transientes [49].	38
Figura 2.21: Arquitectura da plataforma de monitorização e controlo da Universidade Americana de Sharjah, Emirados Árabes Unidos [55].	42
Figura 2.22: Sistema de supervisão e controlo através da Internet da Universidade de Kongu, Índia; adaptado de [18].	43
Figura 2.23: Linha de montagem automatizada [23].	44
Figura 2.24: Arquitectura 3 do sistema de diagnóstico remoto de um processo controlado com PLCs da Universidade do Texas A&M, EUA; adaptado de [23].	45
Figura 3.1: Fluxograma da relação entre controladores e indicadores no diagrama de blocos	49
Figura 3.2: LabVIEW e a rede OPC com o módulo DSC, adaptado de [61].	50
Figura 3.3: Placa DAQ NI USB-6008 do sistema proposto.	51
Figura 3.4: Diagrama da disposição dos pinos da placa DAQ NI USB-6008 [63].	52
Figura 3.5: Arquitectura da rede do sistema de detecção e diagnóstico de falhas proposto.	56
Figura 3.6: Estrutura do <i>website</i> do LabAPI – ISEL.	57
Figura 3.7: Organograma da arquitectura do sistema de detecção e diagnóstico de falhas proposto testado no LabAPI – ISEL.	61
Figura 4.1: Configuração das etiquetas OPC.	64
Figura 4.2: Janela de interface OPC Quick Client.	64
Figura 4.3: Biblioteca de variáveis partilhadas e etiqueta de variável partilhada do PLC “172” no diagrama de blocos.	65
Figura 4.4: VI principal “FDI” e os sub-VIs dependentes no projecto LabVIEW do sistema FDI proposto.	66
Figura 4.5: Diagrama de blocos do sub-VI de autenticação do utilizador.	67
Figura 4.6: Painel frontal do sub-VI de autenticação do utilizador.	68

Figura 4.7: Chamada dos sub-VIs de autenticação do utilizador, e permitir e não permitir no diagrama de blocos do VI principal. _____	70
Figura 4.8: Sub-VI permitir e não permitir no diagrama de blocos. _____	70
Figura 4.9: Experiência “Pórtico”. _____	71
Figura 4.10: Árvore de falhas com lógica booleana para a detecção e diagnóstico de falhas do processo electroíman. _____	73
Figura 4.11: Diagrama de blocos do sub-VI para alarmes do processo electroíman. _____	75
Figura 4.12: Painel frontal do alarme de cor e texto do processo electroíman. _	76
Figura 4.13: Chamada do sub-VI para alarmes do processo electroíman no VI principal do sistema FDI. _____	77
Figura 4.14: Diagrama de blocos do sub-VI <i>Elapsed Time</i> . _____	78
Figura 4.15: Árvore de falhas com lógica booleana para a detecção e diagnóstico de falhas do processo de movimento do pórtico. _____	79
Figura 4.16: Diagrama de blocos do sub-VI para alarmes do processo de movimento do pórtico. _____	79
Figura 4.17: Chamada dos sub-VIs <i>Elapsed Time</i> e alarmes do processo avanço no VI principal do sistema FDI. _____	80
Figura 4.18: Experiência “Cilindros”. _____	81
Figura 4.19: Diagrama de blocos do processo de detecção e diagnóstico de falhas da pressão da rede pneumática. _____	83
Figura 4.20: Diagrama de blocos do sub-VI <i>Elapsed Time_cilindros</i> . _____	85
Figura 4.21: Árvore de falhas com lógica booleana para a detecção e diagnóstico de falhas do processo cilindro não avança/não recua. _____	86
Figura 4.22: Chamada dos sub-VIs <i>Elapsed Time_Cilindros</i> e alarmes do processo avanço e recuo no VI principal do sistema FDI. _____	87
Figura 4.23: Árvore de falhas com lógica booleana para a detecção e diagnóstico de falhas do processo ciclo 1. _____	88
Figura 4.24: Chamada dos sub-VIs <i>Elapsed Time_Cilindros</i> e alarmes do processo ciclo 1 no VI principal do sistema FDI. _____	88

Figura 4.25: Esquema da configuração do sistema de análise de vibrações. ____	89
Figura 4.26: Fluxograma da estrutura do módulo de configuração dos dados de aquisição. _____	90
Figura 4.27: Simulação de sinal através de um <i>express</i> VI no diagrama de blocos. _____	91
Figura 4.28: Aquisição de sinal através do <i>express</i> VI <i>Acquire Sound</i> no diagrama de blocos. _____	92
Figura 4.29: Painel frontal do módulo de configuração dos dados de aquisição do sistema de análise de vibrações em modo simulado _____	93
Figura 4.30: <i>Property Nodes</i> , no diagrama de blocos, para a visibilidade/invisibilidade dos diversos gráficos. _____	94
Figura 4.31: Painel frontal do módulo sinal no tempo. _____	94
Figura 4.32: Estruturas Case das conversões de sinal no diagrama de blocos. ____	95
Figura 4.33: Configuração do filtro <i>Highpass</i> (passa-alto) no diagrama de blocos. _____	96
Figura 4.34: Diagrama de blocos do módulo sinal no tempo. _____	97
Figura 4.35: Painel frontal do módulo espectro FFT. _____	98
Figura 4.36: Configuração do <i>express</i> VI <i>Spectral Measurements</i> no diagrama de blocos. _____	99
Figura 4.37: Diagrama de blocos do sub-VI “rms” pré-alterações. _____	100
Figura 4.38: Diagrama de blocos do sub-VI “rms” pós-alterações. _____	101
Figura 4.39: Diagrama de blocos do sub-VI “pico e rms”. _____	102
Figura 4.40: <i>Bundle</i> de elementos no sub-VI “pico e rms”. _____	103
Figura 4.41: Diagrama de blocos do módulo espectro FFT. _____	104
Figura 4.42: Configuração do filtro anti- <i>aliasing</i> aplicado ao sinal do factor de crista. _____	105
Figura 4.43: Painel frontal do módulo factor de crista. _____	106
Figura 4.44: Diagrama de blocos do módulo factor de crista. _____	106
Figura 4.45: Painel frontal do módulo nível global. _____	107

Lista de Tabelas

Tabela 3.1: Descrição dos sinais da placa DAQ NI USB-6008; adaptado de [63].	53
Tabela 3.2: Falhas e natureza de falhas da experiência “Pórtico”.	59
Tabela 3.3: Falhas e natureza de falhas da experiência “Cilindros”.	60
Tabela 4.1: Propriedades dos PLCs Siemens S7-1200 das experiências “Cilindros” e “Pórtico”.	63
Tabela 4.2: Variáveis lógicas da experiência “Pórtico”.	72
Tabela 4.3: Variáveis lógicas da experiência “Cilindros”.	82
Tabela 4.4: Tabela de severidade da norma ISO 10816-1	108

Lista de Símbolos

f_P	Falhas de processo
f_A	Falhas de actuadores
f_S	Falhas de sensores
f_i	Falhas de origem interna
f_e	Falhas de origem externa
f	Falha
ΔF	Falha de um processo genérico
t	Variável tempo
Y	Variável resultante de um processo com falhas
Y_u	Variável Y_u
U	Variável U
a	Processo sujeito a falhas
Δa	Falha do processo a
$U(t)$	Variável U em função do tempo t ; medida de entrada do processo
S_{ai}	Sintomas analíticos
S_{hi}	Sintomas heurísticos
$Y(t)$	Variável Y em função do tempo t ; medida de saída do processo

r	Resíduos gerados pelos métodos de detecção de falha
θ	Estimativas de parâmetros
\hat{x}	Estimativas de estado
Y_{min}	Limite mínimo da variável de saída Y
$Y_{máx}$	Limite máximo da variável de saída Y
\dot{Y}	Derivada da variável de saída Y
\dot{Y}_{min}	Limite mínimo da derivada da variável de saída Y
$\dot{Y}_{máx}$	Limite máximo da derivada da variável de saída Y
$\frac{dY(t)}{dt}$	Taxa de variação da variável Y em função do tempo t
S^T	Vector de características analíticas e heurísticas
F^T	Vector de falhas
S_n	Vector de referência nominal
F_j	Elementos binários de F
F	Falhas
S	Sintomas
S_i	Entradas ou factos da condição
E_k	Eventos
Z	Amplitude de deslocamento

ω	Frequência
α	Fase
z	Deslocamento
\dot{z}	Velocidade
\ddot{z}	Aceleração
\dot{Z}	Amplitude de velocidade
\ddot{Z}	Amplitude de aceleração
T	Período
N	Número de pontos espaçados e consecutivos
$d\omega$	Resolução em frequência
$\omega_{máx}$	Frequência máxima
ω_a	Frequência de amostragem

1. Introdução

O presente capítulo tem como objectivo realizar uma introdução do trabalho desenvolvido ao longo do Trabalho Final de Mestrado. Na secção 1.1. é apresentada a motivação para o desenvolvimento do tema em estudo. De seguida, na secção 1.2, são expostos os objectivos do presente Trabalho Final de Mestrado. A secção 1.3. descreve de forma sucinta o sistema de diagnóstico de falhas via Internet proposto. Por último, a estrutura do documento é descrita na secção 1.4.

1.1. Motivação

Hoje em dia, observa-se uma evolução muito rápida e significativa da automação industrial devido ao seu desenvolvimento em campos como a computação, a inteligência artificial, e as tecnologias de comunicação. Não só sistemas muito complexos e sofisticados, bem como sistemas simples são completamente automatizados. Os sistemas automatizados são usualmente controlados por Controladores Lógicos Programáveis (Programmable Logic Controller – PLC), e na maioria das vezes, são instalações altamente individualizadas com muitos controladores digitais com entradas e saídas (Inputs and Outputs – I/Os). Assim, com o aumento do número de componentes em processos industriais controlados por autómatos programáveis, a probabilidade de ocorrência de falhas e avarias também aumenta [1].

As empresas industriais têm um interesse crescente em aumentar a disponibilidade dos seus sistemas de produção, principalmente por razões económicas. É aqui, que os conceitos de diagnóstico e isolamento de falhas (Fault Detection and Isolation – FDI) contribuem significativamente para este esforço [1]. No entanto, com a globalização, os fabricantes de equipamentos muitas vezes vendem e instalam equipamento para clientes noutros países. Deste modo, nem sempre a experiência e o conhecimento do processo, por parte dos técnicos, está de imediato disponível no local de fabrico/montagem. Torna-se, por isso, preponderante implementar um sistema que automaticamente detecte e diagnostique falhas remotamente, isto é, um sistema que tenta manter a estabilidade e desempenho do processo quando sujeito a variações no ambiente em que se encontra.

Engenharia remota, manutenção remota ou telemanutenção são palavras-chave para este tipo de engenharia e aplicações de gestão, onde a construção, e a manutenção de instalações e máquinas podem ser monitorizadas e geridas a partir da Internet [2, 3]. Com a Internet cada vez mais indispensável para a distribuição de informação em todo o mundo, surge a noção de Internet das Coisas na Indústria (Industrial Internet of Things – IIoT) e de Indústria 4.0. Estes dois conceitos têm essencialmente uma relação de causa-efeito. Isto é, a IIoT é a base para, e irá resultar na quarta revolução industrial apelidada de Indústria 4.0. Deste modo, a partir da informação da IIoT rapidamente podem ser tomadas acções correctivas, evitando a repetição de erros [4, 5].

Presentemente, os sistemas de Supervisão e Aquisição de Dados (Supervisory Control and Data Acquisition – SCADA) são um exemplo de sistemas de IIoT, sendo largamente utilizados na indústria. Os sistemas SCADA melhoram a eficiência do processo de monitorização e controlo, disponibilizando em tempo útil o estado actual do sistema, através de um conjunto de previsões, gráficos e relatórios, de modo a permitir a tomada de decisões operacionais apropriadas, quer automaticamente, quer por iniciativa do operador [6].

Por outro lado, há mais de duas décadas que instalações de experiências remotas têm estado acessíveis através da Internet. Em 1994, uma das primeiras experiências *online* permitia o controlo remoto do robot ASEA-Irb-6. Os utilizadores podiam controlar a pinça do braço do robot de forma a manipular blocos de madeira numa mesa. O servidor lia os comandos do utilizador introduzidos em HTML, executava a operação requerida e devolvia a página com uma imagem actualizada da instalação [7].

Actualmente, o uso da Internet está disseminado, além de que é um serviço razoavelmente barato, e está disponível em quase todos os estabelecimentos de ensino e empresas. Inclusivamente, a engenharia remota tem-se tornado um elemento importante do ensino em engenharia, levando à necessidade crescente de novos meios de comunicação e de novas ferramentas de aprendizagem, surgindo assim os laboratórios remotos. O objectivo de um laboratório remoto é fazer com que a interacção do aluno com o sistema que se encontra distante seja o mais semelhante possível do trabalho real, no equipamento do laboratório físico, permitindo um uso mais eficiente do equipamento do laboratório. Além do mais,

os laboratórios remotos dão aos utilizadores a oportunidade de conduzir experiências através do conforto de um *browser* na Internet.

Porém, o equipamento físico além de ser acedido por via remota, também pode ser acedido localmente. Por este motivo, o equipamento necessita ser robusto à manipulação descuidada por parte dos alunos. Por vezes, os utilizadores de um laboratório físico têm tendência a danificar, intencionalmente ou não, o material de laboratório. Adicionalmente, devem ser tomadas em consideração medidas de segurança para proteger os utilizadores do laboratório remoto de falhas do sistema. Questões como “O que acontece quando uma das partes móveis do sistema em experiência é subitamente desligado?” têm que ser respondidas antes de garantir acesso à experiência remota. A robustez do equipamento relativamente a falhas ou a indisponibilidade é um assunto chave para a aceitação do paradigma das experiências remotas pelos alunos. Se quando se conectam, não conseguem aceder à experiência escolhida por falha no equipamento, podem perder o interesse [7]. A manutenção de laboratórios remotos é uma tarefa difícil e demorada, quando se pretende que os mesmos estejam disponíveis 24h/dia, 7 dias por semana. O primeiro passo para providenciar uma grande disponibilidade é detectar problemas, o que implica que o equipamento físico e o programa associado sejam capazes de fazer um auto-diagnóstico. Contudo, se a experiência remota não é capaz de voltar a um estado estável conhecido, deve enviar um alarme para o administrador e/ou utilizador [7].

Vastamente utilizados na indústria, os sistemas SCADA estão também a assumir especial importância em laboratórios experimentais para controlo de sistemas auxiliares como distribuição de energia, ventilação, refrigeração, entre outros [6].

Assim, constitui como motivação deste Trabalho Final de Mestrado, o desenvolvimento de um sistema de diagnóstico de falhas via Internet de processos industriais controlados com autómatos programáveis, testado num laboratório remoto e com aplicabilidade em processos industriais diversos. Este tipo de solução é preponderante para que os inconvenientes inerentes à distância entre o utilizador e o equipamento físico sejam minimizados.

1.2. Objectivos

No presente Trabalho Final de Mestrado pretende-se desenvolver e implementar um sistema de diagnóstico de falhas via Internet de processos industriais controlados por autómatos programáveis SIEMENS S7-1200, tendo como caso de estudo o Laboratório Remoto de Automação de Processos Industriais do ISEL (LabAPI). Assim, os objectivos deste Trabalho Final de Mestrado são os seguintes:

- ✓ Criação de um sistema de detecção e diagnóstico de falhas via Internet com aplicabilidade em diversos processos presentes na indústria.
- ✓ Implementação de uma base de dados com o armazenamento dos dados adquiridos e do histórico gerado.

O primeiro objectivo foi atingido a partir da aprendizagem e do aprofundar de conhecimentos de FDI, bem como dos conhecimentos de programação gráfica e via *web*.

Contudo, o segundo objectivo não foi possível implementar no tempo estabelecido para a execução do Trabalho Final de Mestrado, constando como proposta para trabalho futuro na secção 5.2. Trabalho Futuro, página 112.

1.3. O Sistema Proposto de Diagnóstico de Falhas Via Internet

Tendo em conta os recursos existentes no laboratório de Automação Industrial presente na Área Departamental de Engenharia Mecânica, ISEL, e que a maioria das variáveis a monitorizar nos processos industriais são, normalmente, lógicas e a minoria analógicas, definiu-se o desenvolvimento de um sistema de detecção e diagnóstico de falhas via Internet assente no método de verificação de limites fixos e no raciocínio causa-efeito tradicional expresso na forma de árvores de falha de lógica booleana e regras IF-THEN. A verificação de limites é realizada através da verificação das variáveis lógicas de entrada e saída dos processos industriais testados (experiência “Pórtico” e experiência “Cilindros”) no LabAPI – ISEL (laboratório remoto de automação de processos do ISEL, para mais informação consultar Martins [8] e Matias [9]).

Nos processos industriais, muitas máquinas contêm sistemas de accionamento com motores, embraiagens, engrenagens, veios, correias ou correntes, e rolamentos. Estes elementos geram vibrações inerentes às oscilações da máquina, oscilações dos veios, entre outras. Algumas destas vibrações indicam o estado nominal da máquina contudo, o registo de alterações nas mesmas e o aparecimento de vibrações adicionais podem ser originados por falhas. Assim, monitorizar as características de vibração de uma máquina serve para obter informação acerca do estado de “saúde” da mesma e essa informação é utilizada para detectar possíveis falhas e avarias, podendo evitar danos mais graves ou mesmo avarias que possam inutilizar o equipamento e/ou colocar em risco a segurança dos operadores e/ou utilizadores do equipamento.

Dado que, hoje em dia o mundo industrial dá cada vez mais importância à disponibilidade dos seus sistemas de produção, principalmente por razões económicas, a análise de vibrações tem também um papel preponderante no campo da supervisão de máquinas ou de monitorização de um processo industrial [10]. Por este motivo, previu-se a implementação de um sistema de monitorização e análise de vibrações. Suplementarmente e sabendo que o sistema de monitorização e análise de vibrações foi testado num laboratório remoto com o objectivo educacional, estabeleceu-se que seria proveitoso para o utilizador não só poder analisar os parâmetros do sinal adquirido, mas também poder simular um sinal através da escolha de alguns elementos que caracterizam os sinais harmónicos.

1.4. Estrutura do Documento

O presente Trabalho Final de Mestrado encontra-se dividido em cinco capítulos. Além deste primeiro capítulo introdutório, onde se pretende enquadrar o trabalho desenvolvido, este documento é constituído por capítulo 2. Estado da Arte, capítulo 3. Sistema de Detecção e Diagnóstico de Falhas Via Internet Proposto, capítulo 4. Resultados Experimentais e capítulo 5. Conclusão. Estes são sumariamente descritos nos parágrafos seguintes.

No capítulo dois é realizado um levantamento do estado da arte nas áreas de estudo em que este Trabalho Final de Mestrado se insere – diagnóstico de falhas via Internet de processos industriais. Na primeira secção deste capítulo são discutidas as premissas dos actuais desafios da indústria, nomeadamente com referência à

próxima revolução industrial – Indústria 4.0 – aliada à IIoT. Na segunda secção são definidos conceitos como supervisão, falha, avaria e erro, com um breve apanhado pela classificação de falhas. A terceira secção aborda a detecção e o diagnóstico de falhas e os métodos de detecção e de diagnóstico de falhas com particular ênfase nos métodos utilizados no desenvolvimento do sistema proposto. Na quarta secção é desenvolvido o tema da monitorização e análise de vibrações, a sua aplicabilidade e importância na indústria, bem como a descrição de algumas técnicas utilizadas. O capítulo termina com alguns exemplos de sistemas de detecção e diagnóstico de falhas via Internet já existentes.

No terceiro capítulo é apresentado em pormenor o sistema de detecção e diagnóstico de falhas via Internet proposto, sendo descrito detalhadamente a sua arquitectura, programas e linguagens de programação utilizados. A metodologia seguida e as características que compõem o sistema desenvolvido, são descritas no capítulo quatro.

O quinto e último capítulo deste documento expõe as conclusões retiradas ao longo da realização deste Trabalho Final de Mestrado. Adicionalmente, também são sugeridas algumas ideias e melhorias para trabalho futuro.

2. Estado da Arte

O capítulo 2. apresenta um levantamento do estado da arte nas áreas de estudo em que este Trabalho Final de Mestrado se insere – diagnóstico de falhas via Internet de processos industriais. Na primeira secção deste capítulo são discutidas as premissas dos actuais desafios da indústria, nomeadamente com referência à próxima revolução industrial – Indústria 4.0 – aliada à Internet das Coisas na Indústria. Na segunda secção são definidos conceitos como supervisão, falha, avaria e erro, com um breve apanhado pela classificação de falhas. A terceira secção aborda a detecção e o diagnóstico de falhas e os métodos utilizados no desenvolvimento do sistema proposto. Na quarta secção é desenvolvido o tema da monitorização e análise de vibrações, a sua aplicabilidade e importância na indústria, bem como a descrição de algumas técnicas utilizadas. O capítulo termina com alguns exemplos de sistemas de detecção e diagnóstico de falhas via Internet já existentes.

2.1. Indústria 4.0

Ao longo da história, a humanidade atravessou três revoluções industriais. Como se verifica na Figura 2.1, a industrialização começou no final do século XVIII com a mecanização da indústria através da introdução de máquinas a vapor. À primeira revolução industrial seguiu-se, no início do século XX, a produção em massa com o aparecimento da electricidade e das linhas de montagem, originando a segunda revolução industrial. Por último, no final do século XX surge a automação de processos industriais fortemente potenciada pelos componentes electrónicos e as tecnologias de informação (Information Technologies – ITs) dando lugar à terceira revolução industrial [11, 12].

Actualmente, a automação industrial está a atravessar um crescente desenvolvimento com a Internet cada vez mais indispensável para a distribuição de informação em todo o mundo, sendo livre de limitações de tempo, e sendo capaz de transmitir informação numérica e gráfica em qualquer plataforma. Numa rede de automação, à medida que o número de componentes aumenta, a gestão e os problemas de diagnóstico tornam-se cada vez mais desafiantes para a automação e para os profissionais das ITs. Esta evolução leva ao conceito de Internet das

Coisas na Indústria, que pode ser definida como a conectividade inteligente (rede global baseada em protocolos de comunicação) dos equipamentos inteligentes (objectos físicos e virtuais, informação e interfaces). O crescimento da IIoT aliada à consequente reconfiguração e evolução da indústria faz com que provavelmente a quarta revolução industrial – Indústria 4.0 – esteja muito perto de se concretizar (ver Figura 2.1). Pode dizer-se que a Indústria 4.0 é a introdução das tecnologias da Internet na indústria, e a IIoT a motivação e o caminho para a modernização da indústria. A quarta revolução industrial é apelidada de Indústria 4.0 numa alusão à designação do lançamento de novas versões de programas [5, 11, 12, 13].

Desde a Indústria 1.0 até à Indústria 4.0

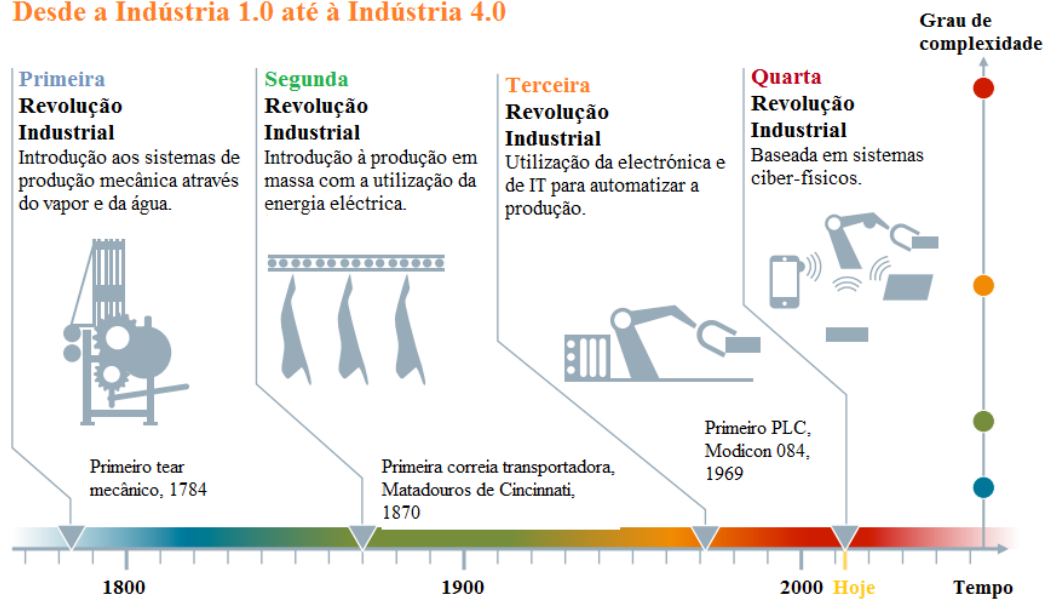


Figura 2.1: Evolução da indústria até à Indústria 4.0; adaptado de [14].

A Indústria 4.0 baseia-se no termo emergente de Sistemas Ciber-Físicos (Cyber-Physical Systems – CPS). Nas últimas décadas verificou-se o desenvolvimento de sistemas integrados, contudo os CPS representam explicitamente um foco na integração da computação com os processos físicos. No futuro representam o próximo passo revolucionário dos sistemas integrados: em conjunto com a Internet e os serviços, e os dados existentes *online*, os sistemas integrados juntam-se para formar os CPS. Geralmente, os CPS são hoje concebidos como uma rede de interacção entre o mundo cibernético e os elementos físicos. [12, 15, 16]. Para mais informações acerca deste tema consultar MacDougall [12] e Leitão *et al.* [16].

Por outro lado, Indústria 4.0 é também o conceito adoptado em empresas, sindicatos de trabalhadores, universidades e agentes governamentais na Alemanha.

Esta iniciativa “Indústria 4.0” representa a visão do país para o futuro da produção industrial, não só na Alemanha como no resto do mundo [12, 13].

Com o apoio das tecnologias emergentes a IIoT é capaz de capturar informação ininterruptamente a partir de vários sensores e objectos, enviar de forma segura valores medidos por sensores para centros de bases de dados, e ajustar continuamente parâmetros de produção através de um sistema de circuito fechado. A Internet das Coisas na Indústria pode detectar eficazmente falhas e desencadear processos de manutenção remotamente. Esta capacidade gerou o conceito de “supervisão e controlo de sistemas através da *web*”, que permite que os utilizadores finais possam aceder a dados em tempo real e controlar equipamentos através de um *web browser*. Os sistemas de Supervisão e Aquisição de Dados representam um exemplo de sistemas no conceito de IIoT. Deste modo, a automação baseada em Internet torna-se possível com o uso de PLCs, sistemas de Supervisão e Aquisição de Dados, ligações de rede virtuais privadas (Virtual Private Network – VPN) e outros elementos de rede, como a Ethernet [5, 6, 17].

Hoje em dia, os sistemas SCADA utilizam tecnologias de computação e comunicação para automatizar a monitorização e controlo dos processos industriais – sistema de supervisão – efectuando recolha de dados em ambientes complexos, eventualmente dispersos geograficamente, e a respectiva apresentação de modo amigável para o utilizador, com recurso a interfaces Humano-Máquina (HMI). Os Sistemas de Supervisão e Aquisição de Dados revelam-se, cada vez mais, de crucial importância na estrutura de gestão das empresas, facto pelo qual deixaram de ser vistos como meras ferramentas operacionais, ou de engenharia, e passaram a ser considerados como uma importante fonte de informação, sendo largamente utilizados na indústria [5].

A Figura 2.2 representa um exemplo da arquitectura de um sistema de supervisão e controlo remoto via internet.

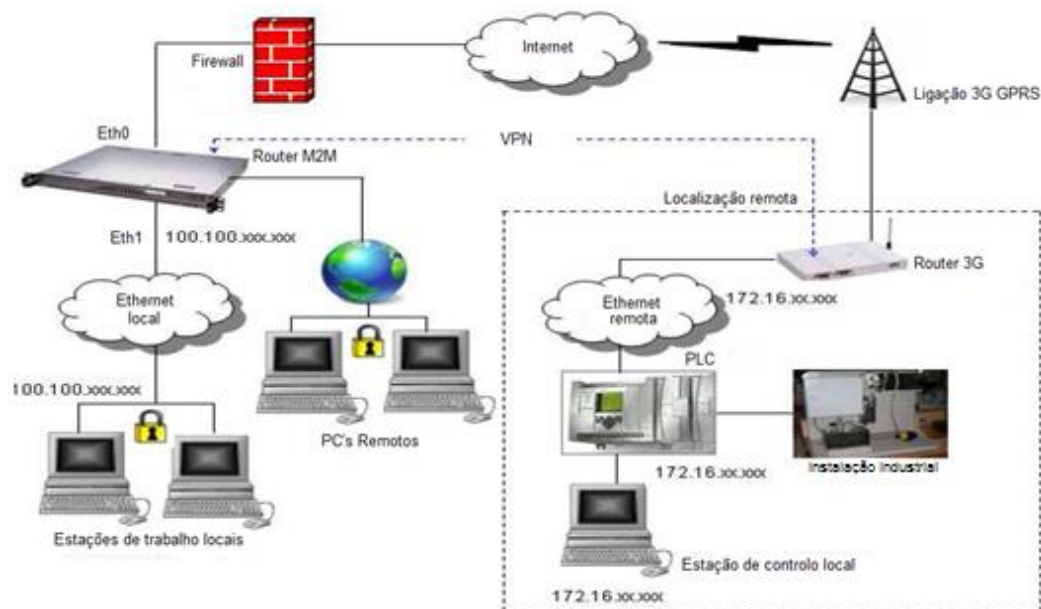


Figura 2.2: Arquitectura do sistema SCADA do Departamento de Engenharia Mecatrónica da Faculdade de Engenharia de Kongu, Índia; adaptado de [18].

Olhando para a estrutura global de um sistema SCADA, existem quatro níveis/conjuntos de elementos distintos dentro deste [19]:

- i. Instrumentação de campo;
- ii. PLCs e/ou Unidades Terminais Remotas (Remote Terminal Units) RTUs;
- iii. Redes de comunicação (usualmente rede Ethernet);
- iv. HMIs - Programa hospedeiro do sistema SCADA.

Este conjunto de elementos tem que providenciar as funções básicas de um sistema SCADA, que são a aquisição de dados, a visualização de dados, o processamento de alarmes e a tolerância de falhas. A função de aquisição de dados envolve a aquisição, transmissão e o armazenamento de dados desde a instalação industrial até ao ponto de monitorização remota. O modo de comunicação para a transmissão de dados é executado por *polling*¹ ou por interrupção². A visualização dos dados consiste na apresentação dos dados previamente adquiridos através de HMIs, geralmente acompanhados por animações com o objectivo de simular a evolução do estado do sistema a monitorizar. Um sistema SCADA deve fazer o processamento de alarmes de forma a informar falhas e/ou avarias verificadas, podendo sugerir medidas de encaminhamento do sistema para um estado de

¹ *Polling* refere-se à activa amostragem do estado de um dispositivo externo através de um programa de cliente, como uma actividade sincronizada.

² Se não existe alteração no estado do sistema, não existe actualização dos dados. Isto é, a actualização só é executada quando existem alterações do estado nominal do processo.

segurança, ou automaticamente intervir mediante as características obtidas. Os alarmes são classificados por níveis de prioridade em função da sua gravidade, sendo reservada a maior prioridade para os alarmes relacionados com questões de segurança [20].

2.2. Monitorização, Detecção e Diagnóstico de Falhas Via Internet

Para a melhoria da confiabilidade, segurança e eficiência da supervisão, a detecção e diagnóstico de falhas tornou-se cada vez mais importante para muitos dos processos técnicos e industriais, como centrais de energia, centrais químicas, siderúrgicas, entre outros. A detecção incipiente de falhas é um ponto essencial para evitar a deterioração do produto, a degradação do desempenho, danos significativos na saúde de máquinas e humana, ou mesmo a perda de vidas, sendo para isso preponderante implementar um sistema de supervisão e monitorização [21].

As funções de supervisão englobam as estruturas destinadas à detecção e diagnóstico de falhas, validação de referências e o próprio supervisor humano. Num sistema de supervisão é preponderante a existência de um sistema de detecção e diagnóstico de falhas capaz de detectar e isolar alterações no funcionamento do sistema. No nível da supervisão podem distinguir-se as seguintes funções [10]:

- **Monitorização:** variáveis mensuráveis são verificadas com especial atenção para as tolerâncias e alarmes gerados para o operador.
- **Protecção automática:** no caso de estados de processo perigosos, a função de monitorização inicia automaticamente uma contra-acção apropriada.
- **Supervisão com diagnóstico de falhas:** baseada em variáveis medidas, são calculadas características, são gerados sintomas através da detecção de alterações, o diagnóstico de falhas é executado e as decisões para contra-acções são cumpridas.

Resumindo, supervisão é monitorizar um sistema físico e tomar as acções adequadas para manter a operação do mesmo em caso de falhas [22].

2.2.1. Falhas, Avarias e Erros

Ao explorar a literatura verifica-se, imediatamente, que a terminologia no campo da detecção e diagnóstico de falhas, não é muito clara e consistente. Muitas vezes as palavras falha e avaria não têm uma clara distinção entre as duas, o que leva a confusão e conseqüentemente dificulta a tarefa de compreender e de comparar diferentes abordagens. Por esse motivo, o Comité Técnico SAFEPROCESS reuniu as definições aceites. Assim, no presente trabalho final de mestrado os significados para os termos falha, avaria e erro são dados tendo por base as definições apresentadas pelo Comité Técnico SAFEPROCESS [22]:

- **Falha:** Uma falha é um desvio não permitido de pelo menos uma propriedade característica ou parâmetro do sistema, do seu estado aceitável/usual/normal.
- **Avaria:** Uma avaria é uma interrupção permanente da capacidade do sistema para executar uma função requerida sob condições específicas de funcionamento.
- **Erro:** Um erro é um desvio entre um valor medido ou calculado (de uma variável de saída) e o valor verdadeiro, específico ou teoreticamente correcto.

Todavia, podem fazer-se algumas observações no âmbito destas definições. Considerando, o contexto de sistemas automatizados, serviço é definido como o resultado da execução de qualquer tarefa de um dado processo. Uma avaria ocorre quando um recurso, que é um conjunto de entidades como controladores, máquinas, ferramentas e programas, deixa de fornecer o serviço pretendido. Uma ou mais falhas podem originar uma avaria. Um erro ocorre quando uma parte dos recursos atinge um estado indesejado. Uma falha é a causa de um erro, uma sequência de erros ou uma avaria, além de que é um estado dentro do sistema [10, 23]. A avaria de um componente pertencente a um sistema pode originar uma falha nesse sistema.

De forma a evitar/minimizar perigos ou estados indesejáveis inerentes a falhas ocorridas num sistema, é necessário aplicar técnicas de FDI.

2.2.2. Classificação de Falhas

De uma forma geral, uma instalação industrial pode ser descrita pelo conjunto de quatro elementos – processo, actuadores, sensores e controlador – ver Figura 2.3. O processo é definido pelo conjunto dos componentes da instalação industrial (sistema de distribuição de medicamentos, campo de centrais solares, entre outras) cujo funcionamento obedece a um conjunto de dinâmicas de complexidade e linearidade variável, ou não-linearidade, e que caracterizam o seu comportamento. Os actuadores são componentes cuja acção permite modificar o estado do processo podendo ser accionados de forma automática ou manual (cilindros pneumáticos/hidráulicos, válvulas, motores, entre outros). Os sensores são elementos pertencentes ao processo e responsáveis pela aquisição das suas grandezas mensuráveis, permitindo determinar o seu estado de funcionamento (sensores de fim-de-curso, temperatura, pressão, entre outros). O controlador permite controlar o processo (por exemplo, controladores lógicos programáveis). Todos estes elementos estão sujeitos a falhas e perturbações de severidade variável que alteram a dinâmica global do sistema [24, 25].

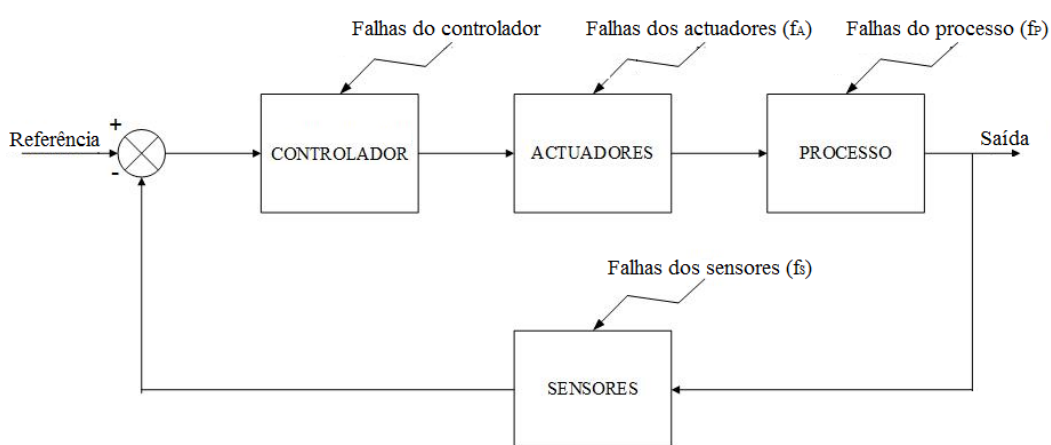


Figura 2.3: Tipos de falhas num sistema.

- **Falhas de Processo (f_P)** – As falhas no processo têm geralmente origem no funcionamento defeituoso dos seus componentes, por exemplo, fugas em tanques, entupimentos de tubagens, entre outros. Este tipo de falhas altera as propriedades dinâmicas das entradas e/ou saídas do sistema.
- **Falhas de Actuadores (f_A)** – As propriedades do sistema não são afectadas, mas a influência do controlador no sistema é interrompido ou modificado.

- **Falhas de Sensores (f_s)** – As propriedades do sistema não são afectadas, mas as leituras do sensor têm erros substanciais.

Também podem existir falhas no controlador, como um erro no PLC, que causam um desvio contínuo ou intermitente do comportamento nominal do processo.

As falhas dos quatro elementos podem ainda ser classificadas tendo em conta a sua origem, natureza temporal e a forma como são modeladas. Deste modo, quanto à sua origem as falhas podem ser organizadas em – Figura 2.4 – falhas internas (f_i) ou falhas externas (f_e) [10]:

- **Falhas de origem Interna (f_i)** – problemas na alimentação de energia, falta de lubrificação e conseqüentemente maior fricção e desgaste, sobreaquecimento, fugas, curto-circuitos.
- **Falhas de origem Externa (f_e)** – influências ambientais como a humidade, poeiras, químicos, radiação electromagnética, temperaturas altas que levam a corrosão, poluição, entre outros.

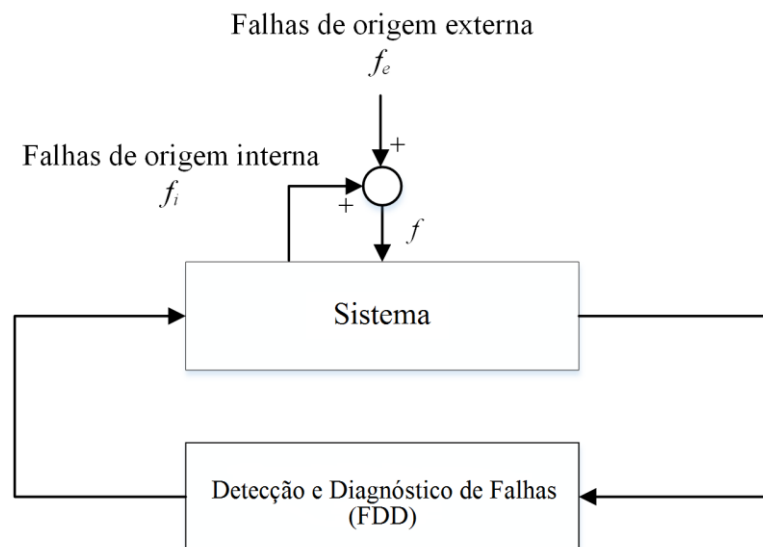


Figura 2.4: Tipo de falhas quanto à sua origem, adaptado de [26].

O conjunto de falha de origem interna (f_i) e falha de origem externa (f_e) constitui uma falha (f) que altera o comportamento nominal de um sistema.

Relativamente à natureza temporal das falhas estas podem desenvolver-se de forma abrupta (passo-a-passo), incipiente ou intermitente, ver Figura 2.5, [27]:

- **Falhas Abruptas** – ocorrem por exemplo em casos de falhas de energia.

- **Falhas Incipientes** – ocorrem por desgaste.
- **Falhas Intermitentes** – um contacto eléctrico intermitente num sistema, por exemplo.

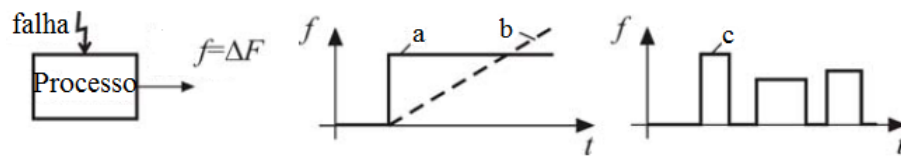


Figura 2.5: Falhas de natureza temporal – a) abruptas; b) incipientes; c) intermitentes; adaptado de [10].

Por outro lado, tendo em conta a detecção de falhas por métodos baseados em modelos (ver secção 2.3.1, página 17), é possível classificar as falhas como falhas aditivas que influenciam a variável Y , ver Figura 2.6 a), pela adição de uma falha f , e falhas multiplicativas pelo produto de outra variável U com falha f , ver Figura 2.6 b). As falhas aditivas aparecem por exemplo em *off-sets* de sensores, enquanto as falhas multiplicativas são alterações de parâmetros dentro do processo.

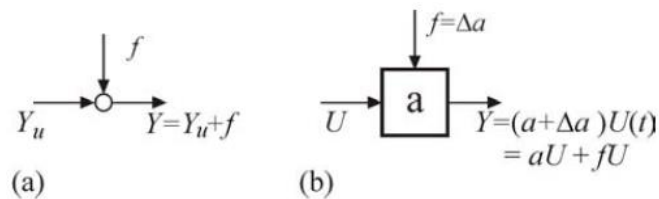


Figura 2.6: Classificação de falhas utilizada em métodos de FDI baseados em modelos de processo – a) aditivas; b) multiplicativas; adaptado de [10].

Existem alguns autores, [28, 29], que também classificam as falhas tendo em consideração a rede. Estas falhas de rede são definidas como [30]:

- **Falhas dos elementos físicos** – são caracterizadas pela inability de entregar correctamente uma informação/serviço. As causas deste tipo de falhas incluem cabos cortados ou a falha de um equipamento de rede essencial como por exemplo, a falha de um *router*.
- **Falhas de programa/programação** – são caracterizadas por uma perda parcial, ou por um atraso na entrega da informação/serviço ou perda da largura de banda de rede. As falhas de programa podem ter como causa a degradação do desempenho como por exemplo o atraso, perda ou redução da largura de banda de rede. Outras podem ter como causa a configuração inadequada da rede (alterações nos parâmetros da rede) e o tráfego novo

gerado por aplicações desconhecidas. O erro também pode estar no controlador.

Nos dois tipos de falhas de rede abordados, o sintoma comum às mesmas é o atraso, este pode ser constante ou variável, importante ou insignificante. Todos os problemas descritos provocam o congestionamento e conseqüentemente aumentam o atraso de forma significativa para as aplicações em tempo-real.

Por último e de uma forma genérica, num sistema automatizado podem ocorrer vários tipos de falhas como por exemplo:

- a falha eléctrica de um sensor que leva a uma medição errada ou a uma medição inexistente;
- a fuga numa válvula, tanque ou tubagem;
- o entupimento de um filtro, válvula ou tubagem;
- a mudança das condições ambientais do processo que causa uma perturbação no mesmo, como a interrupção do fornecimento de ar comprimido necessário para actuar as válvulas pneumáticas;
- a falha de conectividade de dados com alguns equipamentos de controlo.

2.3. Detecção e Diagnóstico de Falhas

O diagnóstico consiste em detectar, isolar e identificar as falhas quando as mesmas ocorrem, determinando a amplitude da falha e o respectivo perfil temporal. A detecção de falhas é uma tarefa cujo objectivo é indicar quando algo corre mal num sistema monitorizado com o máximo de detalhe. O isolamento de falhas consiste em localizar o componente defeituoso. Por último, a identificação de falhas determina a natureza da falha quando esta é detectada [22, 31].

Os sistemas FDI, geralmente, são divididos em dois pontos: o processo automático de leitura e medição de variáveis mensuráveis através de instrumentos – produção de sintomas analíticos; e, observação e avaliação de variáveis e estados por operadores humanos – produção de sintomas heurísticos. Deste modo, o diagnóstico de falhas em máquinas baseia-se em mapear a informação obtida a partir das medições realizadas e/ou das características da máquina que apresenta falhas. Este processo é também denominado por reconhecimento de padrões. Tradicionalmente, o reconhecimento de padrões é realizado manualmente com o

auxílio de ferramentas gráficas como gráficos de espectros. Todavia, o reconhecimento manual requer uma grande experiência por parte dos técnicos na área específica da aplicação em diagnóstico. Assim, o reconhecimento automático torna-se uma ferramenta desejável em diagnóstico de falhas [10, 31].

Sintomas Analíticos

O conhecimento analítico do processo é utilizado na produção de informação analítica quantificável. É executado o processamento de dados das variáveis de processo medidas, para gerar os valores característicos. Isto é, os sintomas analíticos (S_{ai}) são o resultado da verificação de valores limite de sinais mensuráveis, de métodos de detecção de falhas com modelos de processo ou de sinal, e da detecção de alterações [10].

Sintomas Heurísticos

Os sintomas heurísticos (S_{hi}) são as observações do pessoal de operação, na forma de ruído acústico, oscilações ou ilusões ópticas como cores ou fumo, obtidas através de inspeção. Só podem ser representados na forma de medidas qualitativas, por exemplo expressões linguísticas como “pouco”, “médio” ou “muito” [10].

2.3.1. Métodos de Detecção de Falhas

Existem inúmeros métodos de detecção de falhas, no entanto nas secções seguintes apenas são apresentados alguns exemplos com particular detalhe para os métodos utilizados no desenvolvimento do sistema de detecção e diagnóstico de falhas proposto pelo presente Trabalho Final de Mestrado.

A Figura 2.7 representa esquematicamente os vários métodos de detecção de falhas classificados segundo Isermann [10].

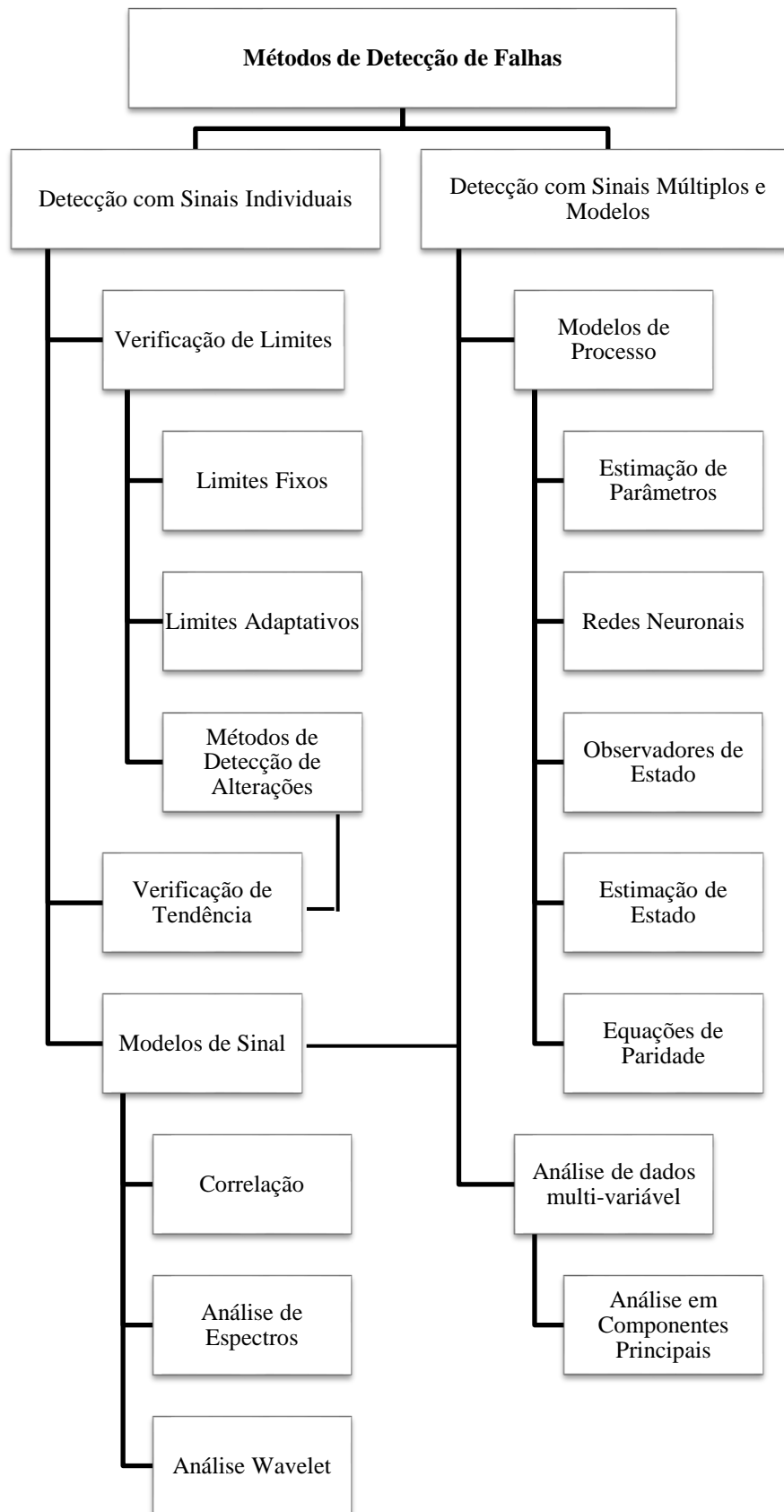


Figura 2.7: Métodos de detecção de falhas; adaptado de [10].

Método de Verificação de Limites

A detecção de falhas com verificação de limites é a abordagem mais clássica, provavelmente a mais primitiva e simples. O método não é baseado em modelos e apenas utiliza a verificação de limites ($Y(t)$) ou verificação da tendência ($\dot{Y}(t)$) de algumas variáveis de saída medidas) [10, 32].

Nesta abordagem, que é largamente utilizada na prática, as medidas do sistema são comparadas, através de computação, com um limite mínimo (Y_{min}) e um limite máximo ($Y_{máx}$) (ver Figura 2.8). O estado nominal do sistema é dado por:

$$Y_{min} < Y(t) < Y_{máx} \quad (2.1)$$

O método de verificação de limites é de fácil interpretação e implementação, e normalmente consiste no disparar de um alarme quando as amostras de sinal saem fora de um determinado limite. Isto é, exceder os limites indica uma situação de alarme. Em muitos sistemas, existem dois níveis de limites, o primeiro limite serve de pré-aviso enquanto o segundo desencadeia uma reacção de emergência.

A verificação de limites pode estender-se ao controlo de tendência de variáveis seleccionadas. Por exemplo, um dos métodos mais conhecidos utiliza a primeira derivada da variável ($\dot{Y}(t) = dY(t)/dt$) e testa se a variação observada está ou não dentro de uma faixa previamente estabelecida (ver Figura 2.8):

$$\dot{Y}_{min} < \dot{Y}(t) < \dot{Y}_{máx} \quad (2.2)$$

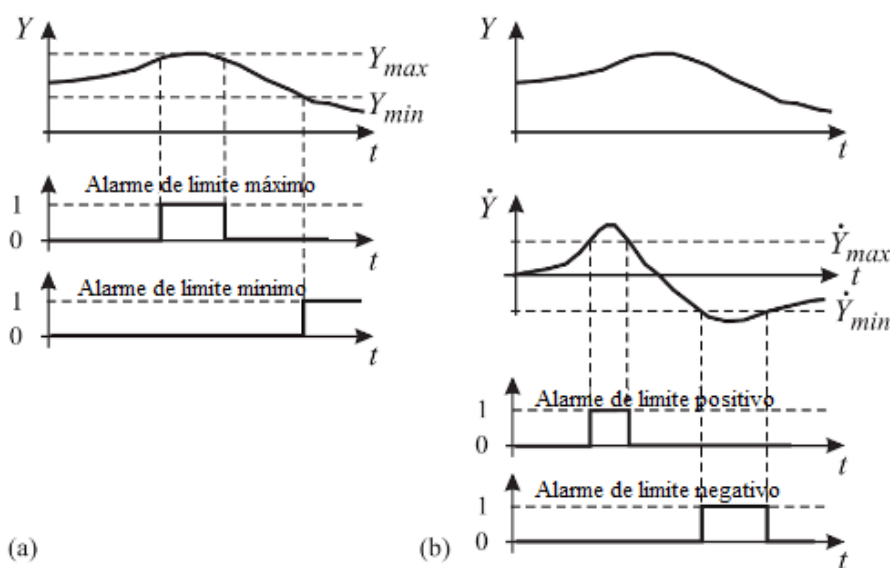


Figura 2.8: Verificação de limites – a) limites fixos; b) limites de tendência; adaptado de [10].

Embora seja simples, o método de verificação de limites sofre de duas desvantagens significativas [33]:

- Como as variáveis do sistema podem sofrer grandes variações, devido a alterações normais das entradas, o teste aos limites tem que ser bastante conservador.
- O efeito de uma falha num único componente pode propagar-se por todas as variáveis do sistema, soltando uma série de alarmes confusos e consequentemente tornando a tarefa de isolar a falha extremamente difícil.

Modelos de Processo

A detecção de falhas com modelos matemáticos consiste na detecção de falhas de processos, de actuadores e/ou de sensores utilizando dependências entre diferentes tipos de sinais mensuráveis. Estas dependências são traduzidas através de modelos matemáticos de processos. Na Figura 2.9 é apresentada a estrutura básica de detecção de falhas baseada em modelos. Através das variáveis de entrada ($U(t)$) e de saída ($Y(t)$) medidas, a detecção de falhas gera resíduos (r), estima parâmetros (θ) ou estados (\hat{x}), e por comparação com os valores nominais detecta sintomas analíticos. A detecção de alterações, tendo as propriedades estatísticas dos resíduos em conta, identificará situações anormais e desencadeará o diagnóstico de falhas que identifica a situação específica de falha [34].

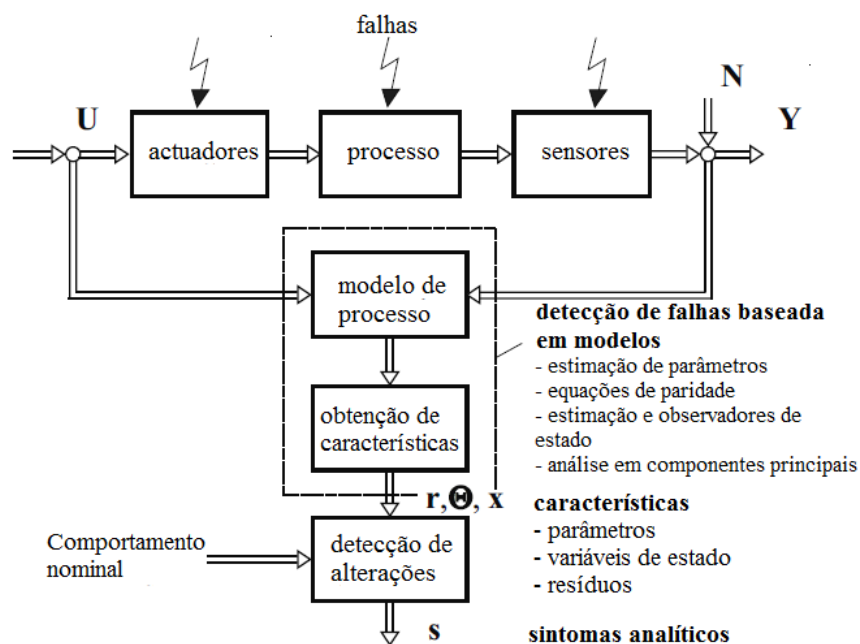


Figura 2.9: Esquema geral do método de detecção de falhas baseado em modelos de processo; adaptado de [10].

Os diferentes tipos de métodos de detecção com modelos de processo estão apresentados no fluxograma da Figura 2.9 e como referência dos mesmos, destacam-se os artigos de Isermann [34] e [35], de Frank [36] e de Gertler [37].

Modelos de Sinal

Na indústria existem muitos processos que se podem caracterizar pelo seu comportamento oscilatório ou cíclico, ao longo do tempo. Nomeadamente, em máquinas rotativas ou correntes alternas. Os sinais resultantes destes equipamentos, são sinais periódicos ou que contêm partes periódicas. Processos aleatórios como o ruído, turbulência ou conduzir sobre superfícies irregulares são sinais estocásticos. Ambos os sinais descritos, podem ser utilizados para detecção de falhas, sendo o objectivo dos métodos baseados em modelos de sinais detectar alterações no comportamento dos sinais provocadas por falhas no processo [10].

Após a medição dos sinais são aplicados modelos matemáticos específicos para por exemplo, obter amplitudes, fases, frequências e funções de correlação para uma banda específica do sinal. Os sintomas analíticos são obtidos através da comparação entre as características medidas e as características do estado nominal do processo. A Figura 2.10 resume o método de detecção de falhas baseado em modelos de sinal [10].

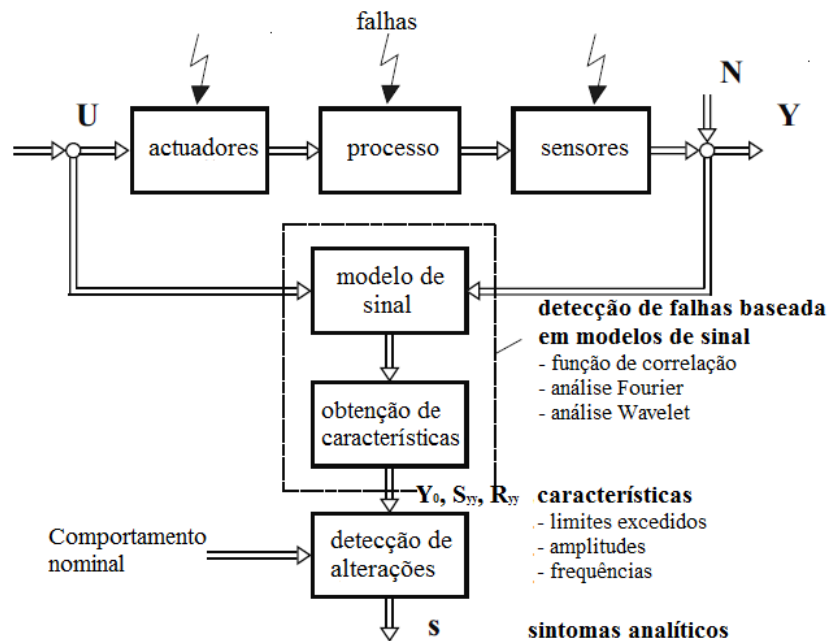


Figura 2.10: Esquema geral do método de detecção de falhas baseado em modelos de sinal; adaptado de [10].

Os métodos de detecção de falhas por modelos de sinal dividem-se em três grandes grupos: sinais periódicos, sinais estocásticos e sinais não estacionários. O filtro passa-banda, a análise do espectro da transformada rápida de Fourier (Fast Fourier Transform – FFT), a estimação de parâmetros espectrais (por exemplo, estimação de amplitudes) e a análise de correlação são exemplos de métodos de detecção de falhas em sinais periódicos. A detecção de falhas em sinais estocásticos é feita com recurso a funções de correlação, análise de espectros e estimação de parâmetros com recurso aos modelos de média móvel auto-regressiva (Autoregressive-Moving Average – ARMA). Por fim, a detecção de falhas de sinais não estacionários pode ser realizada também com recurso a estimação de parâmetros ARMA, análise do espectro da transformada de Fourier por janelas (Short-Time Fourier Transform – STFT) e do espectro da transformada Wavelet [10].

No capítulo 2.4. Monitorização e Análise de Vibrações, página 26, é abordado o método de detecção de falhas baseado em modelos de sinais periódicos com maior detalhe. Para mais informação sobre os restantes métodos consultar Isermann [10].

Um dos métodos de detecção de falhas baseados em modelos de processo é a estimação de parâmetros. Este método pode também ser utilizado em combinação com métodos de detecção de falhas baseadas em modelos de sinal, onde nenhuma entrada é medida (onde apenas se podem medir sinais de saída $y(t)$). Aplica-se especialmente a componentes periódicos onde a estimação de valores característicos (como frequências de ressonância) permitem um conhecimento do estado do processo [38].

2.3.2. Métodos de Diagnóstico de Falhas

Como referido na secção 2.3, o diagnóstico de falhas define-se como o processo de identificar se um sistema está a operar segundo a sua condição de funcionamento nominal ou se está a desviar-se do seu comportamento desejável, determinando o tipo de falha, a sua localização, e quais as potenciais causas para esses comportamentos anormais [2].

Com os resíduos ou características calculados por uma (ou mais) das abordagens descritas anteriormente, o sistema de diagnóstico pode ser conduzido. O método de diagnóstico de falhas pode utilizar o raciocínio causa-efeito tradicional ou executar uma tarefa de classificação. O último é normalmente baseado em

exemplos de referência (classificado por um especialista) e obtidos automaticamente de dados medidos [38]. Para o diagnóstico de falhas, todos os sintomas têm que ser processados de forma a determinar possíveis falhas. Isto pode ser cumprido através de métodos de classificação ou métodos de inferência. Na Figura 2.11 é apresentada uma classificação dos métodos de diagnóstico de falhas adaptada de [10], sendo que existe um grande número de métodos de diagnóstico.

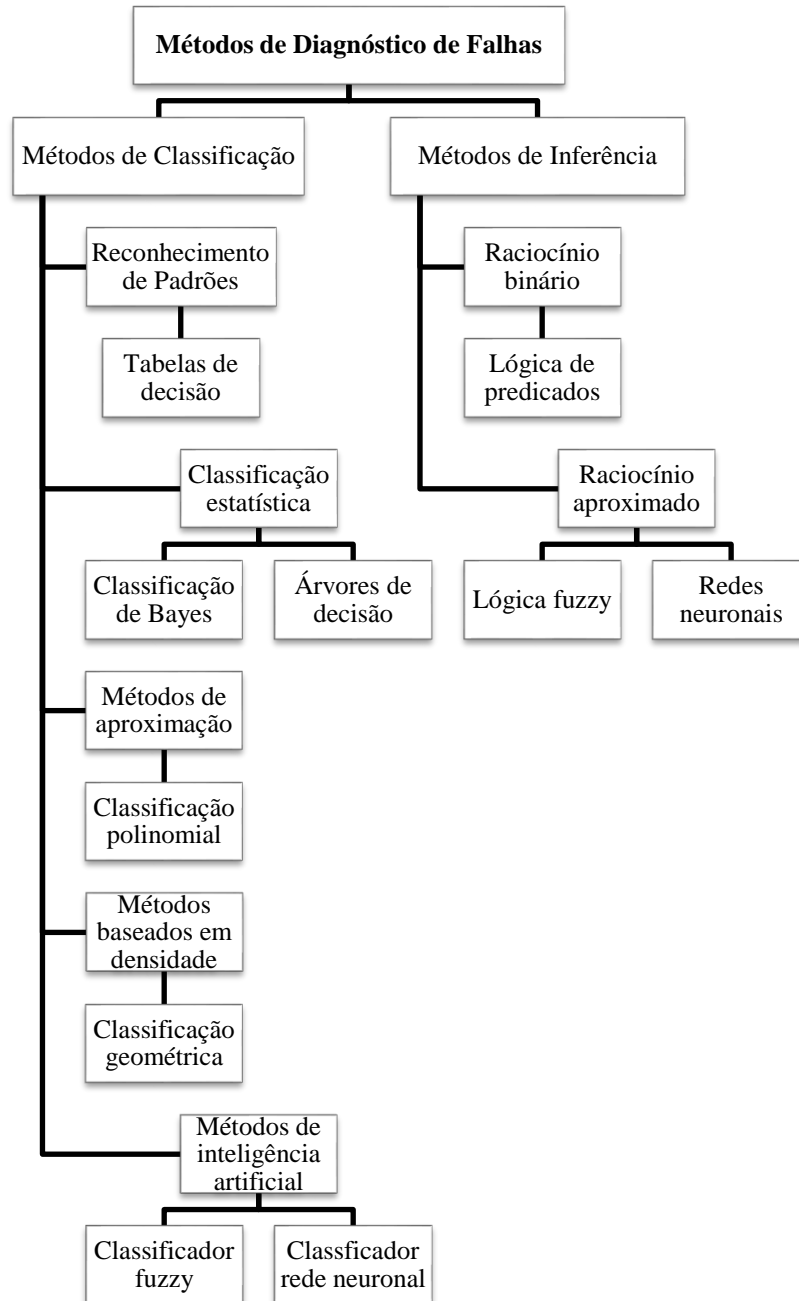


Figura 2.11: Métodos de diagnóstico de falhas adaptado de [10].

Métodos de Classificação

A propagação de falhas de características ou sintomas observáveis segue geralmente uma relação causa-efeito, onde as propriedades e as variáveis físicas estão ligadas entre si quantitativamente bem como em funções de tempo. Contudo, as regras físicas subjacentes frequentemente não são conhecidas numa forma analítica, ou são demasiado complicadas para serem calculadas. Caso não exista informação disponível sobre as relações causa-efeito entre as características e as falhas, os métodos de classificação podem ser aplicados para diagnóstico de falhas. O que conduz a uma base de conhecimento não estruturado. Deste modo, aplicando os métodos de classificação são determinados o vector de referência, S_n , para o comportamento nominal do sistema, e as características analíticas e heurísticas geradas pelos métodos de detecção igualmente em forma de vector [27]:

$$S^T = [S_1, S_2, \dots, S_n] \quad (2.3)$$

As falhas correspondentes, que se assume serem conhecidas [27]:

$$F^T = [F_1, F_2, \dots, F_n] \quad (2.4)$$

Os elementos de F podem ser binários $F_j \in [0,1]$ indicando as falhas como “aconteceram” ou “não aconteceram”. Além disso, podem representar medidas graduais para o tamanho das falhas $F_j \in [0 \dots 1]$.

Portanto, a relação entre F e S é aprendida ou treinada experimentalmente e guardada, constituindo uma base de conhecimento explícito. A partir da comparação entre as características analíticas e heurísticas S^T e a referência nominal S_n , as falhas podem ser diagnosticadas.

Outra opção é usar redes neuronais devido à sua capacidade de aproximar relações não-lineares e de determinar regiões de decisão flexíveis para F em eventos contínuos ou discretos [34].

Métodos de Inferência

Para determinados processos técnicos, as relações básicas entre falhas e sintomas são pelo menos parcialmente conhecidas. Este conhecimento a-priori pode ser representado em termos de relações causa-efeito: falha \rightarrow eventos \rightarrow sintomas.

No caso em que as causalidades dos sintomas de falha possam ser expressas na forma de regras IF-THEN, os métodos de inferência podem ser utilizados.

Na Figura 2.12 a) está representada uma rede causal. As relações aqui presentes seguem o princípio de análise de árvore de falhas (FTA), isto é, partir de falhas através de eventos até aos sintomas ou a análise de árvore de eventos (ETA), partir de sintomas para as falhas.

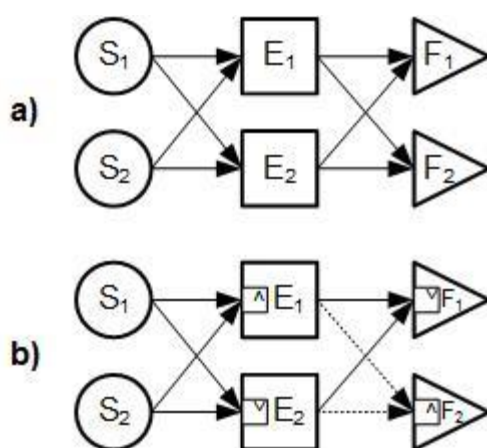


Figura 2.12: Métodos de inferência para o diagnóstico de falhas a) redes causais b) árvores de falha [34].

Para chegar ao diagnóstico, a informação qualitativa (simbólica) pode ser expressa na forma de lógica booleana ou regras IF-THEN de lógica difusa, em que IF <condição> e THEN <conclusão>. Os sintomas S_i são entradas (factos) da condição, enquanto os eventos E_k e as falhas F_j são a causa lógica dos factos. Caso exista mais do que um sintoma para um evento ou falha, os sintomas são associados por meio de AND (\wedge) e OR (\vee):

$$IF \langle S_1 \wedge S_2 \rangle THEN \langle E_1 \rangle$$

$$IF \langle E_1 \vee E_2 \rangle THEN \langle F_1 \rangle$$

Para uma dada representação qualitativa, podem ser utilizadas inúmeras abordagens diferentes para o diagnóstico de falhas. Na secção 2.5, Figura 2.11 é dada uma visão dos vários métodos de diagnóstico de falhas via Inferência, sendo possível aprofundar mais conhecimento sobre os mesmos em Isermann [10] e Venkatasubramanian *et al.* [39].

2.4. Monitorização e Análise de Vibrações

A maioria das pessoas está familiarizada com a vibração pois no dia-a-dia testemunham-se muitos exemplos da mesma, nomeadamente devido à existência de inúmeras actividades humanas que envolvem vibração de uma maneira ou de outra. O Universo e a Vida são movimento e grande parte desse movimento manifesta-se na forma de oscilações ou vibrações. Por exemplo, o movimento dos planetas – actividade geológica – é um movimento oscilatório que pode causar vibrações massivas na forma de terremotos. A fala humana requer movimentos oscilatórios da laringe. Os seres humanos ouvem devido à vibração dos tímpanos e vêem porque as ondas de luz estão sujeitas a vibração. Por sua vez, a respiração está associada à vibração dos pulmões e andar envolve movimentos oscilatórios das pernas e mãos. O balançar de um pêndulo tem um movimento oscilatório e vibratório e a corda de uma guitarra vibra. Deste modo, um objecto que vibra move-se de e para, trás e frente. Um objecto que vibra oscila [40].

Existem formas de vibração que são perceptíveis a “olho nú”, ou seja, é visível o movimento de vai-e-vem de um objecto que vibra. Também é possível tocar em algo a vibrar e sentir a vibração. Outras vezes, a vibração pode criar sons que se ouvem ou calor que se sente. Todavia, em instalações industriais existe um tipo de vibração mais preocupante e na maioria das vezes imperceptível: a vibração mecânica de máquinas. A vibração mecânica ocorre com frequência em vários processos e equipamentos presentes numa instalação industrial. Esta não produz apenas ruído, que pode prejudicar a saúde humana, mas também tem a energia necessária que pode reduzir a vida de um componente/máquina e afectar a estabilidade da operação. Contudo, a vibração de uma máquina pode ser intencionalmente desenvolvida e por isso ter um propósito funcional. Nem todos os tipos de vibração de máquinas são indesejados. Por exemplo, a vibração é útil em peneiras, compactadores, finalizadores de superfícies, escovas de dentes eléctricas, máquinas de lavar, entre outros. Por outro lado e como discutido anteriormente, a vibração pode ser não intencional e pode levar a danos nas máquinas, constituindo, na maioria das vezes, uma vibração não deliberada e indesejada [40, 41].

Monitorizar as características de vibração de uma máquina serve para obter informação acerca do estado de “saúde” da máquina. Essa informação é utilizada

para detectar possíveis falhas e avarias, podendo evitar danos mais graves ou mesmo avarias que possam inutilizar o equipamento e/ou colocar em risco a segurança dos operadores e/ou utilizadores do equipamento. Apesar do diagnóstico de falhas em máquinas englobar a medição e análise de vários fenómenos associados à operação das mesmas, o comportamento vibratório é o principal indicador do estado actual da máquina. Assim, a monitorização e análise de vibrações tem um papel preponderante no diagnóstico de máquinas e na condição actual das mesmas [42, 43, 44].

Habitualmente, um sistema de detecção e diagnóstico de falhas baseado em vibrações consiste em três passos fundamentais (ver Figura 2.13):

1. Aquisição de dados – reunião de dados relevantes para a condição actual da máquina;
2. Processamento de dados – análise dos dados ou sinais adquiridos no passo 1, para melhor compreender ou interpretar os dados;
3. Tomada de decisão – com os resultados obtidos no passo 2 são tomadas decisões quanto ao estado da máquina por parte do operador ou automaticamente por parte do sistema de detecção e diagnóstico de falhas.

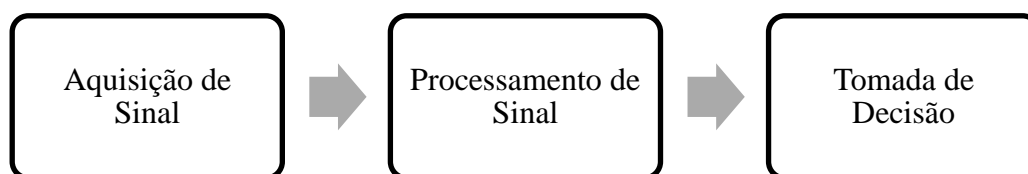


Figura 2.13: Três passos num sistema de detecção e diagnóstico de falhas.

2.4.1. Conceitos de Vibração

Como referido em 2.4. Monitorização e Análise de Vibrações, a maioria dos indivíduos tem um conhecimento intuitivo do que é a vibração, no entanto para ser adquirida e usada como indicador do estado real do equipamento, é importante compreender explicitamente o que é a vibração e toda a problemática envolvente.

A vibração pode definir-se como “uma oscilação em que a quantidade é um parâmetro que define o movimento de um sistema mecânico” [45]. Isto é, a vibração é um movimento que se repete num intervalo de tempo resultado da oscilação de um objecto em relação à sua posição de repouso. A forma mais simples de vibração é a vibração harmónica (ver Figura 2.14), o que faz com que

todas as outras formas de vibração não sejam mais do que a soma de duas ou mais vibrações harmónicas.

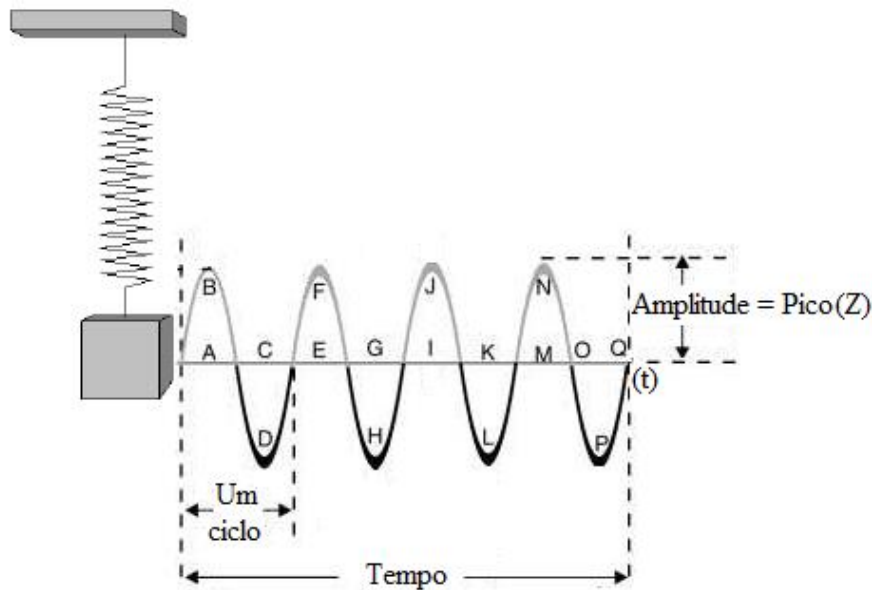


Figura 2.14: Vibração harmónica.

Os termos numéricos mais importantes para descrever a vibração harmónica são:

- a amplitude (Z) – valor de afastamento do ponto ou sistema relativamente à posição de referência, no caso de uma vibração harmónica a amplitude é o valor máximo (pico) da sinusoidal, M-N na Figura 2.14. As vibrações podem ser medidas em deslocamento [μm], velocidade [mm/s] ou aceleração [m/s^2 ou g];
- a frequência ou frequências (ω) – número completo de ciclos por unidade de tempo (segundo), reproduzido na Figura 2.14. Habitualmente, a unidade é o Hertz [Hz], podendo também ser representada em ciclos por minuto [cpm], ou em rotações por minuto [rpm];
- e a fase (α) - a posição inicial do ponto, no instante ($t=0$) em que se começou a medir e é indicada em radianos [rad] ou graus [$^\circ$].

As funções harmónicas são frequentemente utilizadas na análise de vibrações. Um objecto que sofra uma vibração harmónica simples é definida por:

$$z(t) = Z \cos(2\pi\omega t + \alpha) \quad (2.5)$$

Onde ω é a frequência da harmónica, $2\pi\omega t$ a frequência angular correspondente, Z a amplitude de deslocamento e α a fase.

A velocidade $\dot{z}(t)$ e a aceleração $\ddot{z}(t)$ do objecto são obtidas através da derivação do deslocamento uma e duas vezes, respectivamente:

$$\dot{z}(t) = 2\pi\omega Z \sin(2\pi\omega t + \alpha) = \dot{Z} \cos\left(2\pi\omega t + \alpha + \frac{\pi}{2}\right) \quad (2.6)$$

$$\ddot{z}(t) = -(2\pi\omega)^2 Z \cos(2\pi\omega t + \alpha) = \ddot{Z} \cos(2\pi\omega t + \alpha + \pi) \quad (2.7)$$

Assim, os valores absolutos máximos de deslocamento, velocidade, e aceleração de um objecto que apresenta uma vibração harmónica, ocorrem quando as funções trigonométricas de (2.5) a (2.7) são numericamente iguais à unidade. Estes valores são conhecidos como, respectivamente, amplitude de deslocamento, velocidade, e aceleração, sendo definidos matematicamente como:

$$Z = Z \quad (2.8)$$

$$\dot{Z} = (2\pi\omega)Z \quad (2.9)$$

$$\ddot{Z} = (2\pi\omega)\dot{Z} \quad (2.10)$$

As vibrações encontradas na prática muitas vezes não apresentam este padrão regular. Nas máquinas as falhas podem gerar impulsos, ou sinais harmónicos estacionários adicionais. Os sinais harmónicos surgem devido a efeitos linearmente sobrepostos como o desequilíbrio, o alinhamento inexacto, veios deformados, falhas nos dentes de engrenagens, falhas em rolamentos, diferenças no fluxo eléctrico de motores eléctricos, ou por mudanças na operação periódica da máquina [10].

Assim, as vibrações podem ser a combinação de várias ondas sinusoidais, cada uma com amplitude e frequência diferentes, e são denominadas de vibrações periódicas ou determinísticas [45]. As vibrações periódicas são as mais importantes quando se trata de apurar o estado actual das máquinas. A cada ciclo de rotação dá-se a repetição da ocorrência dos fenómenos na máquina. Nas estruturas, as vibrações determinísticas são o resultado da acção de máquinas em funcionamento ao seu redor. A Figura 2.15 exhibe o somatório de várias vibrações harmónicas, cada uma com diferentes amplitudes e diferentes frequências, dando origem a uma vibração periódica, igualmente representadas no domínio do tempo e no domínio da frequência. Um sinal no domínio da frequência dá origem a um gráfico da amplitude de resposta da vibração *versus* a frequência.

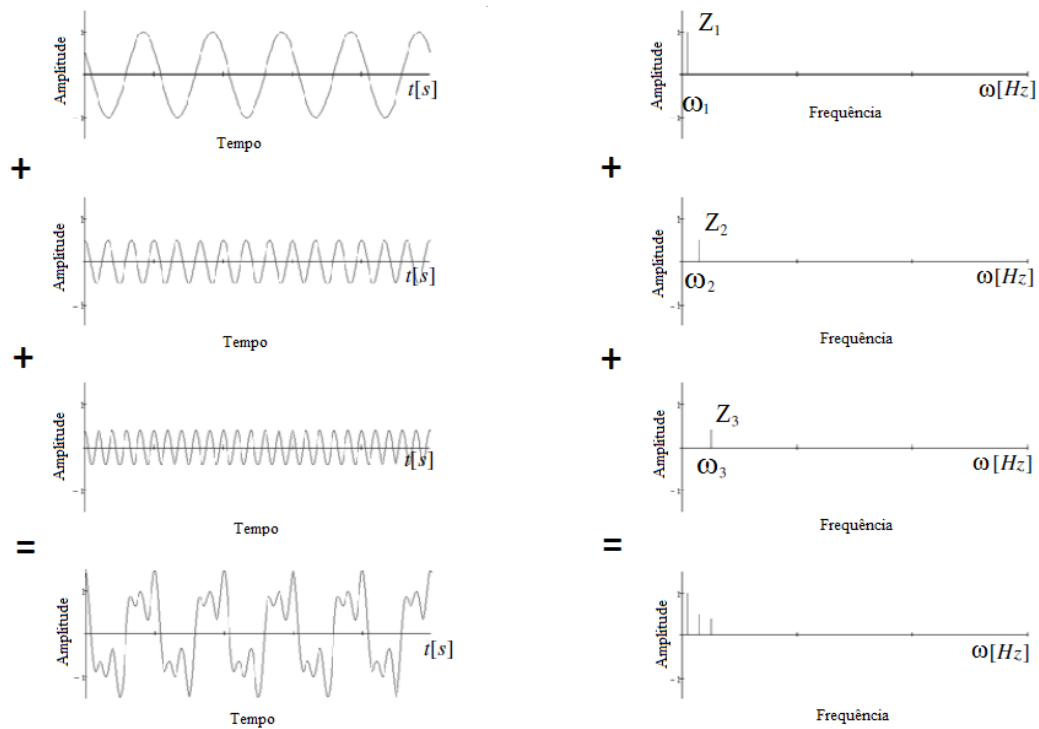


Figura 2.15: Vibração periódica representada no domínio do tempo e na frequência do sinal.

Com base na Figura 2.15 é possível extrair as funções harmônicas, (2.11) a (2.13), e conseqüentemente a função da vibração periódica, (2.14), que resulta da soma das três vibrações harmônicas simples:

$$z_1(t) = Z_1 \cos(2\pi\omega_1 t + \alpha_1) \quad (2.11)$$

$$z_2(t) = Z_2 \cos(2\pi\omega_2 t + \alpha_2) \quad (2.12)$$

$$z_3(t) = Z_3 \cos(2\pi\omega_3 t + \alpha_3) \quad (2.13)$$

$$z(t) = z_1(t) + z_2(t) + z_3(t) \quad (2.14)$$

Conclui-se que a vibração é um fenômeno dinâmico que pode ser representada em deslocamento, velocidade e aceleração, tanto no domínio do tempo como no domínio da frequência.

2.4.2. Aquisição e Processamento de Sinal

A vibração é a fonte básica de sinal que será analisado através de técnicas de processamento de sinal. Contudo, para que se obtenha este sinal é necessário adquiri-lo através de sensores ou transdutores. O sensor é um dispositivo que recebe um estímulo de entrada e responde com um sinal elétrico. O transdutor é um dispositivo que converte um tipo de energia, como a vibração, num diferente

tipo de energia, habitualmente corrente ou tensão eléctrica. Deste modo, muitos sensores podem ser designados de transdutores. Em monitorização de vibrações os sensores mais comuns são sensores de: deslocamento, velocidade e aceleração. Estes são sensores que produzem nos seus terminais de saída uma corrente ou tensão eléctrica que é proporcional, respectivamente, ao/à deslocamento, velocidade e aceleração a que são sujeitos.

Os acelerómetros (sensores de aceleração) piezoeléctricos são os mais utilizados para a medição de vibrações em máquinas porque abrangem uma vasta gama de frequências, são mais pequenos e por isso oferecem globalmente melhores características que outro tipo de sensor [46]. No entanto, como indicado em 2.4.1. Conceitos de Vibração, o deslocamento, a velocidade e aceleração estão relacionadas através de funções simples de derivação ou integração. Assim, independentemente do sensor utilizado (deslocamento, velocidade ou aceleração) para a aquisição, o sinal pode ser processado directamente ou passar por processos matemáticos de conversão para outras unidades de medida de vibrações. Estas operações são geralmente efectuadas através de sistemas de processamento de sinal electrónicos e digitais. Deste modo, cada tipo de sensor tem as suas vantagens dependendo da aplicação, tendo também as suas limitações. Logo, deve considerar-se a especificidade da aplicação e seleccionar o sensor mais adequado para a situação a analisar.

A medição de sinal pode ser realizada de duas formas: em-linha (*online*) ou em-diferido (*offline*). Na medição em-linha, ou aquisição automática, os sensores estão continuamente em utilização. No caso da medição em-diferido, ou aquisição manual, o sistema de aquisição de dados é portátil, ou seja, é feita uma medição intermitente ou periódica. A aquisição de dados é um método eficaz e não destrutivo que permite monitorizar a condição da máquina enquanto esta está no seu funcionamento estacionário ou num estado transitório.

É muito importante que seja feita uma correcta aquisição de sinal, caso contrário o tratamento dos dados será afectado e todo o trabalho de detecção e diagnóstico de falhas estará comprometido.

Como referido anteriormente, a análise de vibrações começa com um sinal real que varia com o tempo, adquirido a partir de um sensor. Através da entrada deste sinal num instrumento de medição de vibrações, existe uma panóplia de opções

possíveis para processar e consequentemente analisar o sinal e comumente, distinguem-se em três categorias principais [31]:

- análise no domínio do tempo – baseia-se na própria forma de onda do sinal adquirido. Calcula características a partir dos sinais no tempo tais como valor de pico, valor de pico-pico, factor de crista, valor de RMS, etc;
- análise no domínio da frequência – baseia-se no sinal transformado no domínio da frequência. A vantagem deste tipo de análise quando comparada com a análise no domínio do tempo é a habilidade de facilmente identificar e isolar componentes de frequência de maior interesse. O método mais comum de processamento de sinal no domínio da frequência é através da aplicação do algoritmo da transformada rápida de Fourier;
- e análise tempo/frequência – baseia-se na análise de sinais em ambos os domínios, tempo e frequência, e tem como vantagem em relação à análise no domínio da frequência a capacidade para lidar com sinais não-estacionários, que são os mais frequentes quando ocorrem falhas em máquinas. Habitualmente, o processo mais utilizado é a STFT.

Além dos exemplos referidos anteriormente, existem outras técnicas relacionadas com as três categoriais discutidas como por exemplo: o sinal no tempo, o nível global, o espectro de frequência PBC (Porcentagem de Banda Constante), a fase e os modos de vibração, a análise de tendência, a média no tempo, a média de espectros, o envelope, o cepstrum, as cascatas, os testes de ressonância, entre outras.

Hoje em dia, com o aumento da disponibilidade dos pacotes de processamento de sinal e com a globalização da tecnologia e Internet, todas as técnicas referidas, e um grande número de outras, podem agora ser directamente experimentadas e/ou programadas pelo utilizador num computador genérico.

Sinal no Tempo

O sinal no tempo permite visualizar a vibração em tempo real e toda a informação está contida neste parâmetro sensível ao dano, sendo por isso o parâmetro mais intuitivo. Todavia, a representação dos fenómenos vibratórios no domínio do tempo torna difícil a sua interpretação. Neste tipo de representação todos os dados

são adicionados para indicar o deslocamento total a qualquer instante sendo, habitualmente, penoso determinar a contribuição de uma fonte particular de vibração. O sinal no tempo acaba por não fornecer a quantidade de informação útil desejada [47].

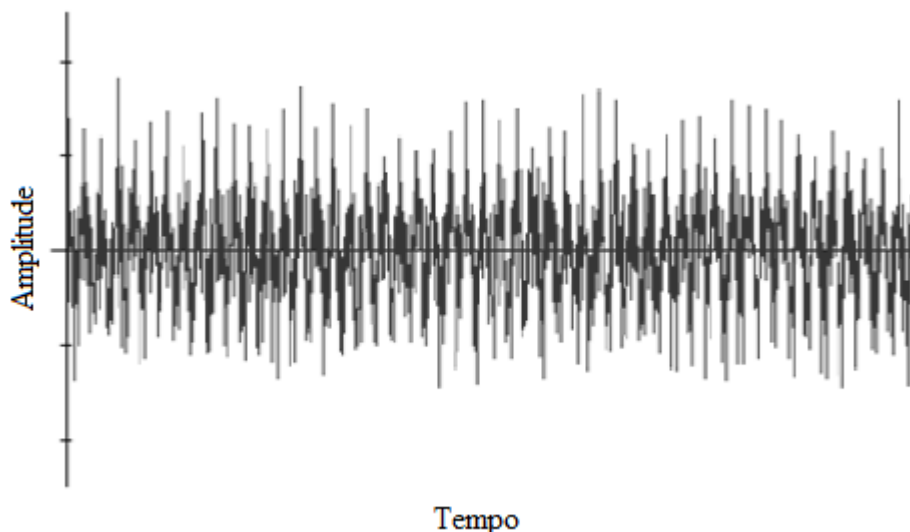


Figura 2.16: Exemplo de sinal no tempo.

2.4.3. Espectro FFT

Actualmente, a forma mais comum de processamento de sinal no domínio do tempo para o domínio da frequência é a transformada rápida de Fourier. A FFT é um algoritmo que calcula a transformada discreta num intervalo de tempo. Este método utiliza o sinal no tempo e decompõe-no nas suas componentes de frequência, determinando quais as harmónicas que compõem a vibração medida e cada harmónica com amplitude, fase e frequência [48]. Este tipo de análise no domínio da frequência através da FFT tem aplicabilidade em sinais periódicos.

A Figura 2.17 mostra um exemplo da conversão de sinal, em que cada linha no espectro de frequência representa uma harmónica do sinal adquirido no domínio do tempo. O intervalo de tempo é definido como N pontos espaçados e consecutivos, do sinal adquirido. Como se verifica na Figura 2.17, este intervalo de tempo é convertido como um bloco completo num bloco completo de linhas de frequência, em que cada linha do espectro FFT representa uma frequência. Todas as amostras do intervalo de tempo são necessárias para produzir cada linha no domínio da frequência. Sendo importante ressaltar que, num analisador dinâmico

de sinal não se obtém um resultado válido até estar reunido um intervalo de tempo completo.

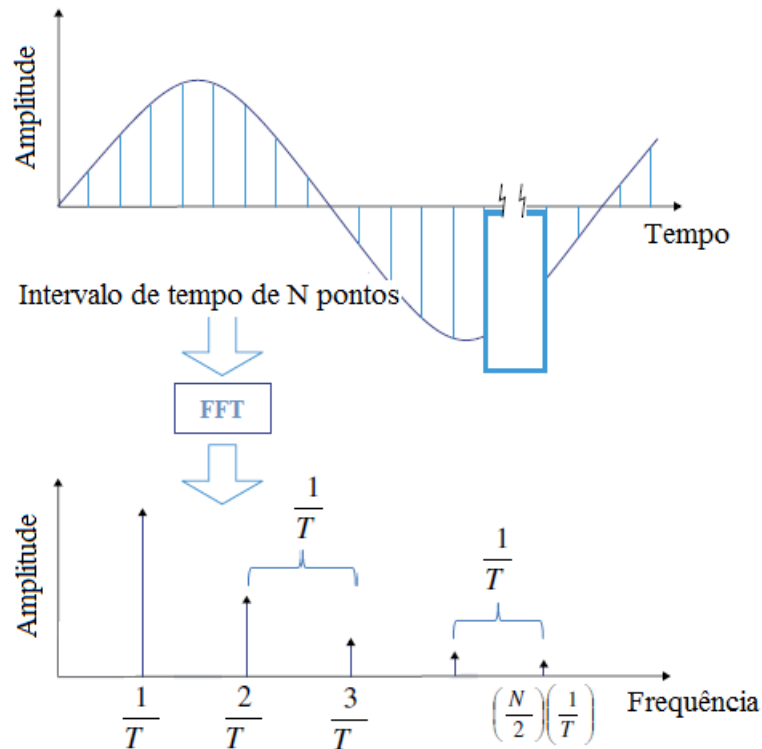


Figura 2.17: Sinal representado no domínio do tempo e no domínio da frequência; adaptado de [49].

Outra característica da FFT é que transforma estas amostras no domínio do tempo em $N/2$ linhas de frequência, igualmente espaçadas de $1/T$ (em que T é o período de amostragem). Isto é, só se obtém metade das linhas porque na verdade, cada frequência contém duas fracções de informação, amplitude e fase [48]. Portanto, é favorável que o número de pontos a medir (N) seja uma potência de 2, por exemplo 1024. A primeira frequência do espectro de frequência é $1/T$ [Hz] e a última é $(N/2)(1/T)$ [Hz]. O espaçamento entre linhas de frequência é denominado de resolução em frequência ($d\omega$). Com estes dados, é então possível definir o valor de frequência mais alto que pode ser adquirido, designado de frequência máxima ($\omega_{máx}$). Esta é dada por (2.15):

$$\omega_{máx} = \frac{N}{2} \times \frac{1}{T} \quad (2.15)$$

Em que N é o número de pontos a medir.

Aliasing

Para uma análise adequada do espectro FFT é indispensável obter um número elevado de amostras por segundo – frequência de amostragem (ω_a) elevada – devido a um problema denominado *Aliasing*. O *Aliasing* é um potencial problema de qualquer sistema de dados de amostragem, e genericamente faz com que frequências altas após a amostragem aparentem ser frequências baixas, o que poderá esconder dados importantes para o diagnóstico de falhas. De forma a evitar o fenómeno de *Aliasing*, o teorema de Claude Shannon e Harry Nyquist (Critério de Nyquist) mostra que para definir um sinal correctamente, a frequência de amostragem deve ser no mínimo o dobro da frequência máxima (ver (2.16)) [48, 50]:

$$\omega_a = 2 \times \omega_{m\acute{a}x} \quad (2.16)$$

Deste modo, é preponderante aplicar ao espectro um filtro anti-*aliasing* para exceder no dobro o conteúdo da frequência mais alta. Normalmente, o valor adequado e empregado é de 2,56 vezes o valor de $\omega_{m\acute{a}x}$ [48].

Escala de Amplitudes e Médias na Frequência

Frequentemente, é possível definir a escala de amplitudes de um espectro de frequência FFT, que pode ser linear ou logarítmica. A escala linear adequa-se quando se pretende realçar as maiores amplitudes presentes no espectro. Porém, a escala logarítmica é a mais usual porque permite não só a visualização das amplitudes maiores bem como das mais pequenas, no mesmo gráfico. Logo, quando existem, no mesmo espectro, amplitudes muito grandes e amplitudes muito pequenas deve utilizar-se a escala logarítmica caso contrário corre-se o risco de não observar as pequenas.

Dado que, no mundo real o sinal adquirido não é limpo, isto é, é adquirido na presença de ruído significativo, ao espectro FFT podem ser aplicadas médias que têm como função atenuar ou retirar os efeitos de ruído e/ou de componentes aleatórias do sinal monitorizado, tornando o espectro mais consistente e consequentemente realçando as frequências que mais interessam. Quanto maior o número de médias, mais perto o ruído está de zero o que significa que mais limpo será o espectro. Existem diversos tipos de médias das quais se destacam a média de ponderação linear e a média de ponderação exponencial. A média linear define

que todos os espectros têm o mesmo peso, enquanto a média exponencial define que o espectro mais recente tem mais peso que os demais. A primeira é útil para a obtenção de dados com repetibilidade. A última é vantajosa para observar condições que variam pouco ao longo da frequência de amostragem – processo estacionário [48, 51].

Janelas

Existe um outro problema decorrente da utilização da transformada rápida de Fourier para processamento de sinal – o *leakage* ou efeito de janela. *Leakage* define-se com a inexistência de um número inteiro de ciclos do sinal amostrado, provocando o aparecimento de falsas componentes de frequência no espectro FFT. Observando a Figura 2.18 a) compreende-se que o espectro de frequência deste sinal no tempo tem apenas uma linha (uma frequência). Contudo e porque apenas se extrai um intervalo de tempo finito (ver Figura 2.18 b)), o efeito de janela ocorre e a onda será distorcida (ver Figura 2.18 c)) dando origem a um espectro FFT diferente.

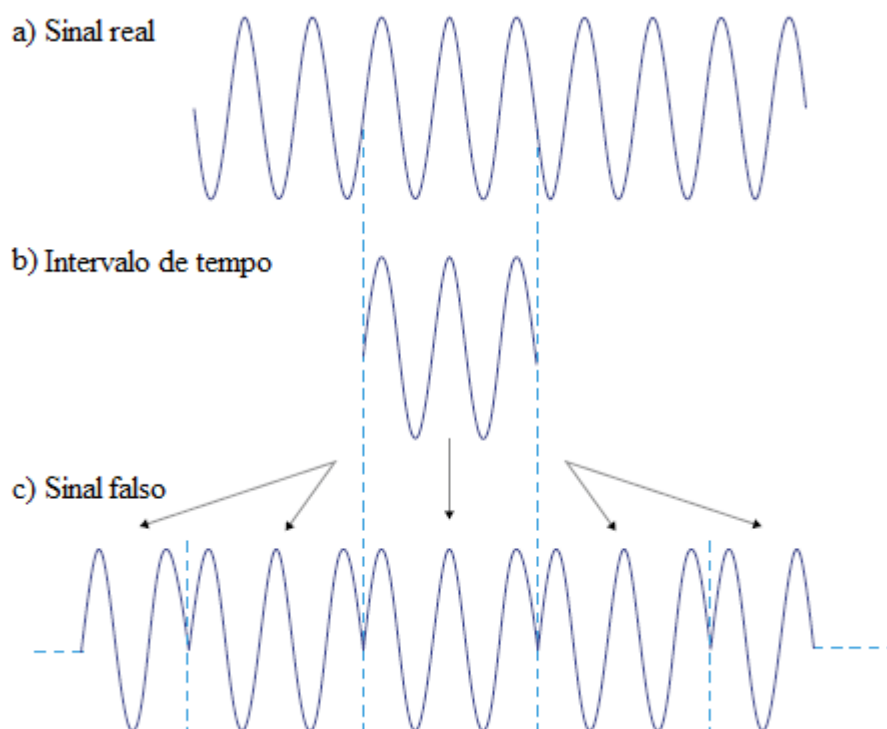


Figura 2.18: Exemplo do efeito de janela num sinal não-periódico no intervalo de tempo [49].

A solução para esta questão passa por aplicar uma janela apropriada que irá minimizar o erro do efeito de janela. A janela “obriga” a FFT a olhar para os dados

através de uma janela estreita onde a entrada é zero em ambas as extremidades do intervalo de tempo [49]. Exemplificando, na Figura 2.19 c) está representado um espectro FFT que traduz, de forma aproximada, o sinal no domínio da frequência apresentado no domínio do tempo (ver Figura 2.19 a)). De referir, que se diz de forma aproximada pois a janela reduz o *leakage* mas não o elimina na sua totalidade. A Figura 2.19 b) exhibe o fenómeno de *leakage* quando aplicado o algoritmo FFT num sinal não-periódico, em que a FFT vê as descontinuidades como uma frequência de modulação.

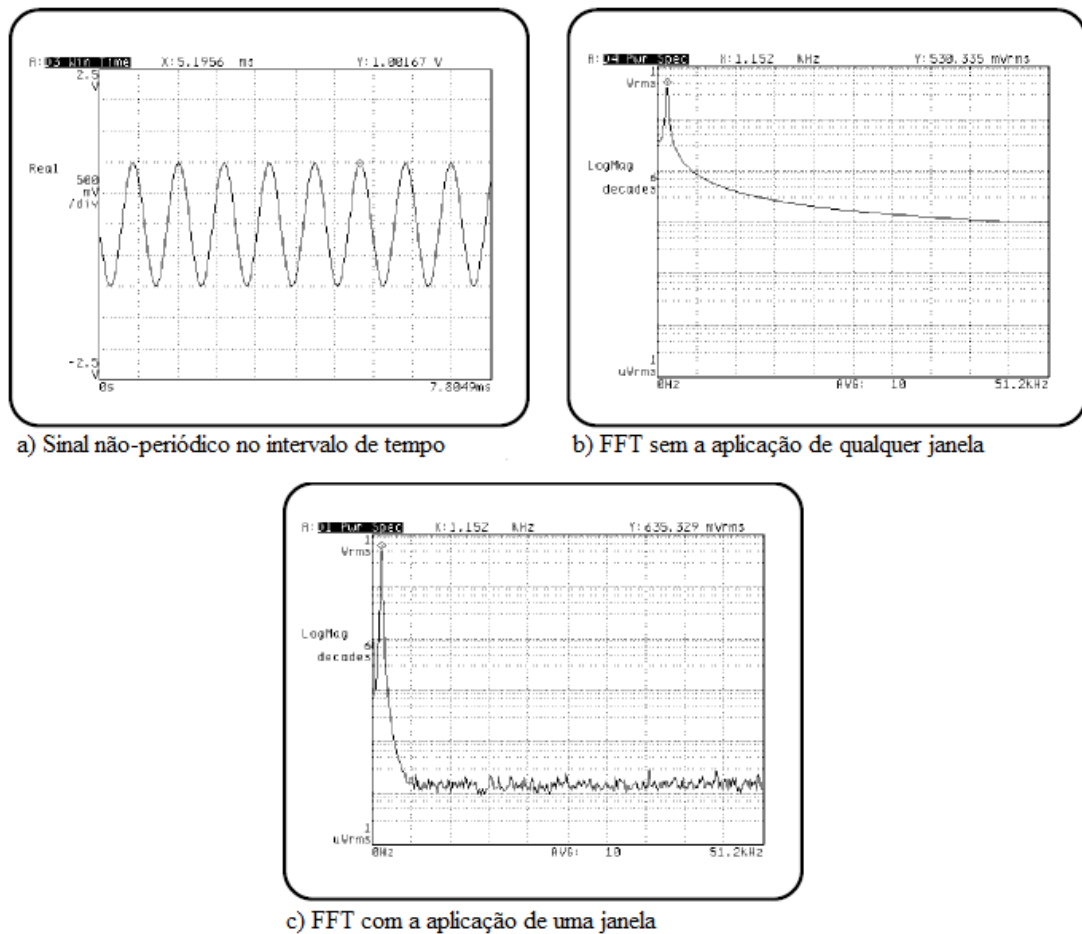


Figura 2.19: Atenuação do fenómeno de *leakage* com a aplicação de janela apropriada (adaptado de [49]).

A janela mais utilizada em análise de vibrações de sinais estacionários é a *Hanning*, pois oferece uma boa relação entre resolução de frequência e resolução de amplitude, com amplitudes relativamente exactas. Porém, para o regime transiente existem outras janelas mais adequadas, como a janela Rectangular, visto que a *Hanning* irá retirar o regime transiente ao sinal, isto é, existirá perda de informação. A janela Rectangular, ao contrário da *Hanning*, pesa uniformemente todo o intervalo de tempo [48, 49]. A Figura 2.20 mostra a diferença entre utilizar,

num regime transiente, a janela *Hanning* (ver Figura 2.20 b)) e a janela Rectangular (ver Figura 2.20 c)).

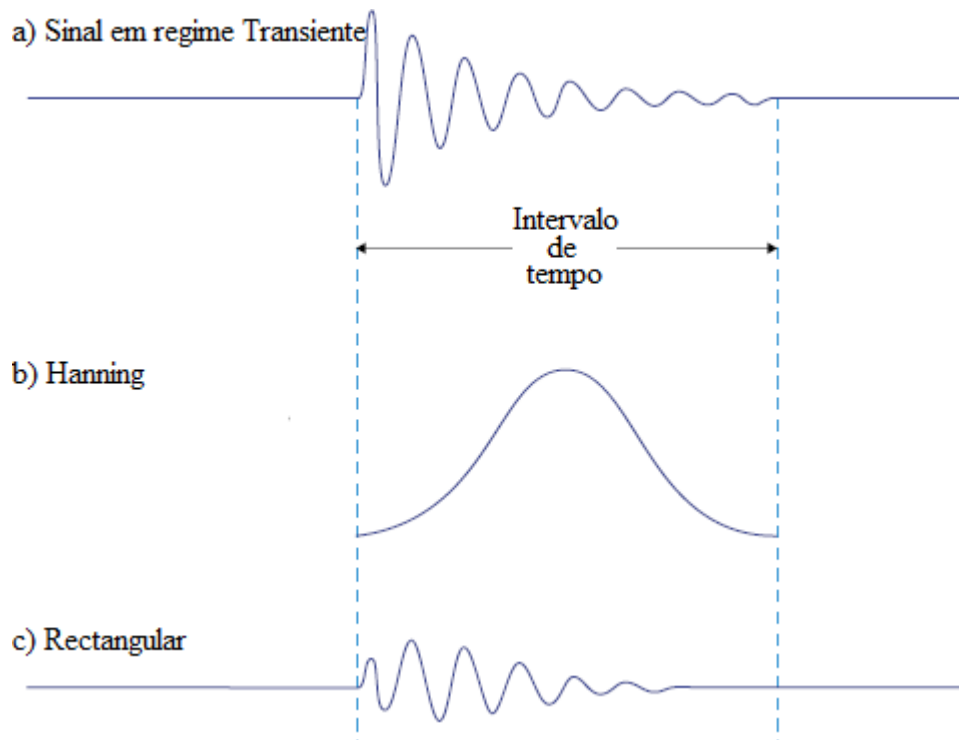


Figura 2.20: Aplicação da janela Rectangular em regimes transientes [49].

Outros tipos de janelas incluem a:

- *Flat Top* – tem como principal função a calibração de sinais. É uma solução que providencia amplitudes de sinal com maior precisão (minimiza o erro da amplitude) no entanto exibe um sinal que aparece em várias frequências devido ao filtro muito largo [48, 50];
- *Hamming* – oferece melhor resolução em frequência comparada com as demais, mas fraca resolução em amplitude. Esta janela é adequada para separar componentes de frequência adjacentes [48, 49];
- *Blackman-Harris* – este tipo de janela possui características e utilidade semelhantes à *Hamming* [48];
- *Kaiser-Bessel* – é a janela mais selectiva e que provê a melhor separação de componentes de frequência muito próximas, porque apresenta ainda menos *leakage* [48, 50];
- entre outras.

Por último, referir que o espectro FFT é a técnica mais utilizada para diagnóstico de falhas pois permite grandes resoluções em frequência e diagnostica a maior

parte das causas de falha. No entanto, não é uma técnica de detecção e exige conhecimentos mais especializados.

2.4.4. Outras Técnicas de Processamento de Sinal

Como referido na secção 2.4.2. Aquisição e Processamento de Sinal, existe um grande número de opções de processamento e análise de sinal adquirido consoante a aplicação pretendida. De seguida são discutidas de forma breve mais algumas técnicas.

Nível Global

Para certos objectivos da análise de vibrações, é conveniente expressar a amplitude em termos do valor médio da função harmónica, isto é, do valor RMS (*Root Mean Square*). O nível global consiste geralmente na medição da velocidade eficaz ou RMS numa banda entre 10 e 1000 [Hz]. O RMS é calculado a partir do somatório de todos os valores de velocidade elevados ao quadrado, divididos pelo número de medições e seguido da aplicação da raiz quadrada a esta operação (ver (2.17)) [47, 52]:

$$RMS = \sqrt{\frac{\sum_{i=0}^{N-1} \dot{z}_i^2}{N}} \quad (2.17)$$

Em que \dot{z}_i são os valores de velocidade para N números de pontos a adquiridos.

Este parâmetro sensível ao dano também pode ser calculado em unidades de deslocamento e aceleração, seja em pico (amplitude máxima), pico-pico (amplitude dupla) ou RMS, numa banda de frequência ou numa frequência específica. Existem inúmeros tipos de nível global

Dado que, a vibração de máquinas é composta por várias frequências os valores de RMS obtidos são comparados com os respectivos valores de referência ou com tabelas de severidade de vibração de modo a determinar a condição relativa da máquina [47, 52]. Entre as tabelas de severidade mais conhecidas distinguem-se as seguintes:

- ISO 10816 – Vibrações Mecânicas – Avaliação das vibrações de máquinas através de medições em componentes não-rotativas;

- ISO 7919-1986 – Vibração de Máquinas Não-alternativas (deslocamento pico-pico);
- e ISO 2954-1973 – Requisitos para Instrumentos de Medida de Severidade de Vibração.

As tabelas de severidade podem ser utilizadas como guia, contudo caso exista experiência por parte do utilizador deve estabelecer-se os limites de alerta e perigo baseados nessa experiência.

O nível global fornece informação sobre o estado geral da máquina mas não permite identificar as componentes em frequência dominantes, nem tem grande sensibilidade a determinadas alterações dinâmicas, devido sobretudo ao facto de ser uma média. O nível global também detecta avarias mas num estado mais avançado. Ou seja, vai no sentido contrário ao desejável que é uma detecção incipiente [47]. Todavia, é simples e não necessita de um grande investimento visto que pode ser obtido computacionalmente através de um programa adequado e da aplicação de (2.17), ou similar, ao espectro do sinal adquirido. Os níveis de vibração global podem ser traçados num gráfico em função do tempo de forma a apurar quão rapidamente a condição da máquina está a evoluir.

Factor de Crista

No espectro FFT o impacto e o ruído têm a mesma aparência, no entanto no campo da análise de vibrações têm significados diferentes. Caso o utilizador visualize somente o espectro FFT, não irá conseguir diferenciar impactos de ruído ocasional, resultando na perda de informação essencial para a correcta análise do estado actual da/do máquina/componente. Por outro lado e ainda que, a medição da velocidade eficaz seja um bom indicador para detectar danos em aplicações estacionárias, o mesmo não acontece caso o sinal adquirido contenha informação proveniente de mais de um componente, como por exemplo no caso da vibração de uma caixa de velocidades que consiste no conjunto de várias engrenagens, veios e rolamentos. Assim, é também vantajoso obter parâmetros que meçam a característica pontiaguda de um sinal, como o factor de crista [47, 51, 53].

O factor de crista define-se como o quociente entre o pico (positivo ou negativo) e o RMS da vibração e contém informação acerca de ambos os níveis (ver (2.18)):

$$\text{Factor de Crista} = \frac{\text{Pico}}{\text{RMS}} \quad (2.18)$$

Este parâmetro sensível ao dano é significativo apenas onde os valores de pico são razoavelmente uniformes e repetíveis de uma amostra de sinal para outra. Por esta razão, o factor de crista é especialmente adequado para caracterizar sinais que contêm impulsos repetitivos, como a vibração de máquinas alternativas e a vibração produzida por falhas em engrenagens e rolamentos, sinais que se caracterizam por possuírem altas frequências e reduzidos deslocamentos [51].

Concluindo, o factor de crista é um cálculo rápido e bastante útil, porém detém algumas limitações e pode revelar-se ineficaz na identificação de certas falhas. Por exemplo, se a falha ocorre de forma progressiva, o valor de RMS do sinal pode estar a crescer gradualmente, embora o factor de crista apresente uma tendência decrescente.

No sentido de aprofundar mais conhecimento sobre outras técnicas de processamento de sinal em análise de vibrações consultar Rao [40, 47], Randall [50] e Courrech *et al.* [54].

2.5. Trabalho Existente em Detecção e Diagnóstico de Falhas Via Internet

Existem inúmeros exemplos de sistemas de monitorização, detecção e diagnóstico de falhas na indústria e ensino em todo mundo, tendo-se optado por apresentar, nesta secção, as seguintes soluções de diversos sectores da engenharia:

Plataforma de Monitorização e Controlo para Aplicações Controladas Com PLCs (Universidade Americana de Sharjah, Emirados Árabes Unidos)

A plataforma proposta pela Universidade Americana de Sharjah, Emirados Árabes Unidos, teve como objectivo a implementação de um sistema em rede para monitorizar e controlar remotamente PLCs via internet ou via rede GSM. A arquitectura do sistema é apresentada na Figura 2.21.

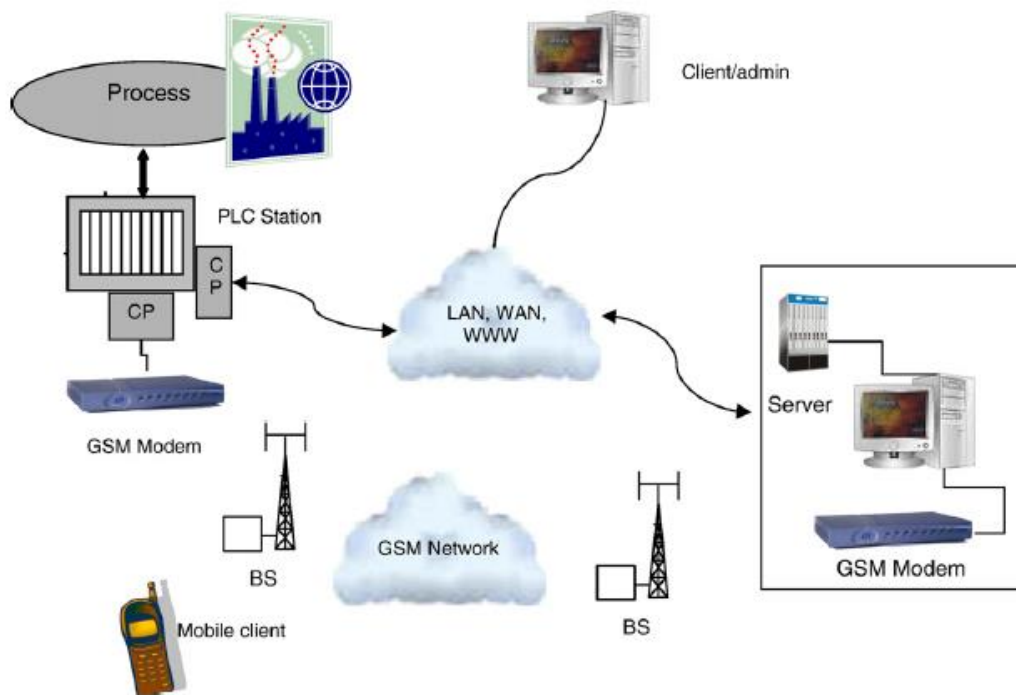


Figura 2.21: Arquitectura da plataforma de monitorização e controlo da Universidade Americano de Sharjah, Emirados Árabes Unidos [55].

A plataforma é construída com recurso aos PLCs Siemens S7. Estes PLCs têm um processador de comunicação integrado que permite ao PLC comunicar via LAN, WAN ou via rede GSM. A monitorização e o controlo dos processos industriais por parte do cliente ou do administrador é feita via rede Ethernet ou *wireless* via Internet. O sistema inclui uma base de dados, no servidor, conectada ao processo via Ethernet [55].

O programa do sistema foi implementado com recurso ao programa Simatic Manager da Siemens e a Java. O Simatic Manager é utilizado para fazer a comunicação com os PLCs e a aplicação Java para estabelecer a comunicação entre a base de dados e a estação de PLCs. A base de dados guarda o estado e a leitura das variáveis de entrada (leitura dos sensores) dos PLCs em tabelas de tempo, e executa, desde que requisitada, qualquer análise de dados. O sistema também recebe e executa ordens do administrador ou dos clientes para controlar o processo. Foi também desenvolvido um mecanismo de alerta de erros que fornece informação importante de diagnóstico ao administrador. O estado actual do processo pode ser comunicado ao administrador através de mensagens escritas (Short Message Service – SMS) via rede GSM [55].

Sistema de Supervisão e Informação Remoto Via Internet (Departamento de Engenharia Mecatrónica da Universidade de Engenharia de Kongu, Índia)

O sistema de supervisão proposto é aplicado no controlo de um processo real no Laboratório de Interface e Automação do Departamento de Engenharia Mecatrónica da Universidade de Engenharia de Kongu, Índia.

O controlo do processo inclui um tanque de aquecimento e dois tanques de água: um no fundo do tanque de aquecimento, que actua como reservatório, e outro no topo que alimenta o tanque de aquecimento. Os PLCs funcionam como um sistema de controlo local. O objectivo do sistema de supervisão e controlo via internet implementado é de manter a relação entre a entrada e a saída do processo quando este é perturbado de forma crítica. O sistema é baseado numa configuração cliente/servidor via Internet (ver Figura 2.22) com troca de informação entre os processos experimentais e os clientes, em que os mesmos podem monitorizar, controlar e deste modo modificar os parâmetros do processo [18].

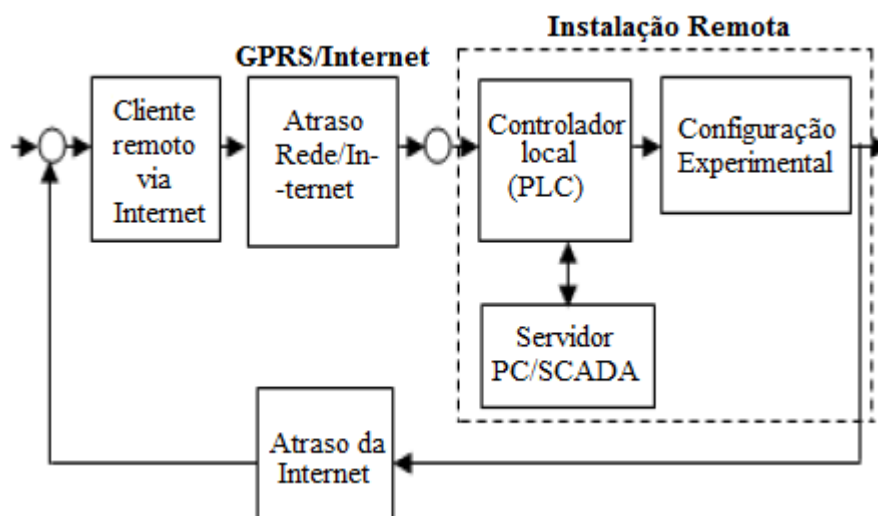


Figura 2.22: Sistema de supervisão e controlo através da Internet da Universidade de Kongu, Índia; adaptado de [18].

A arquitectura proposta (ver Figura 2.2, página 10) para o sistema de supervisão e controlo da Universidade de Kongu, Índia, tem cinco características fundamentais [18]:

- Sem limitações de distância, se estiverem disponíveis as redes de GSM/GPRS-2G/3G;
- Comunicação segura de duas vias (IP estático e IP público);
- Expande a rede Ethernet até à rede GPRS;

- Oferece ao cliente remoto um endereço IP estático independente para conexão remota com os equipamentos de porta de entrada;
- *Firewall* e ligação VPN para uma comunicação segura.

Diagnóstico de Falhas Remoto de Um Processo Controlado com PLCs (Universidade do Texas A&M, EUA)

Neste laboratório foi implementado um sistema de diagnóstico de falhas remoto aplicado num processo controlado com autômatos programáveis como se verifica na Figura 2.23.



Figura 2.23: Linha de montagem automatizada [23].

Foram concebidos três níveis de arquiteturas diferentes, sendo que a mais completa é a arquitectura de nível 3 representada na Figura 2.24.

A arquitectura 1 possui capacidades de conectividade remota e colaboração entre o solucionador de problemas remoto e o operador do sistema. Além disso, as suas capacidades incluem vídeo-conferência (operador local e técnico especializado remoto), imagens instantâneas através de *webcam*, comunicação textual, e transferências seguras de ficheiros. A arquitectura 2 detém as capacidades da arquitectura 1 e adicionalmente permite acesso remoto directo ao PLC, inserção

de sensores suplementares externos ao sistema (célula de carga e sensor de pressão), e monitorização do estado do processo praticamente em tempo-real através de uma interface gráfica executada em LabVIEW. Por fim, a arquitectura 3 engloba as duas arquitecturas anteriormente descritas com características adicionais como uma interface de monitorização hierárquica do processo, reprodução de vídeo, registo de eventos num histórico, e ambiente de trabalho remoto [23].

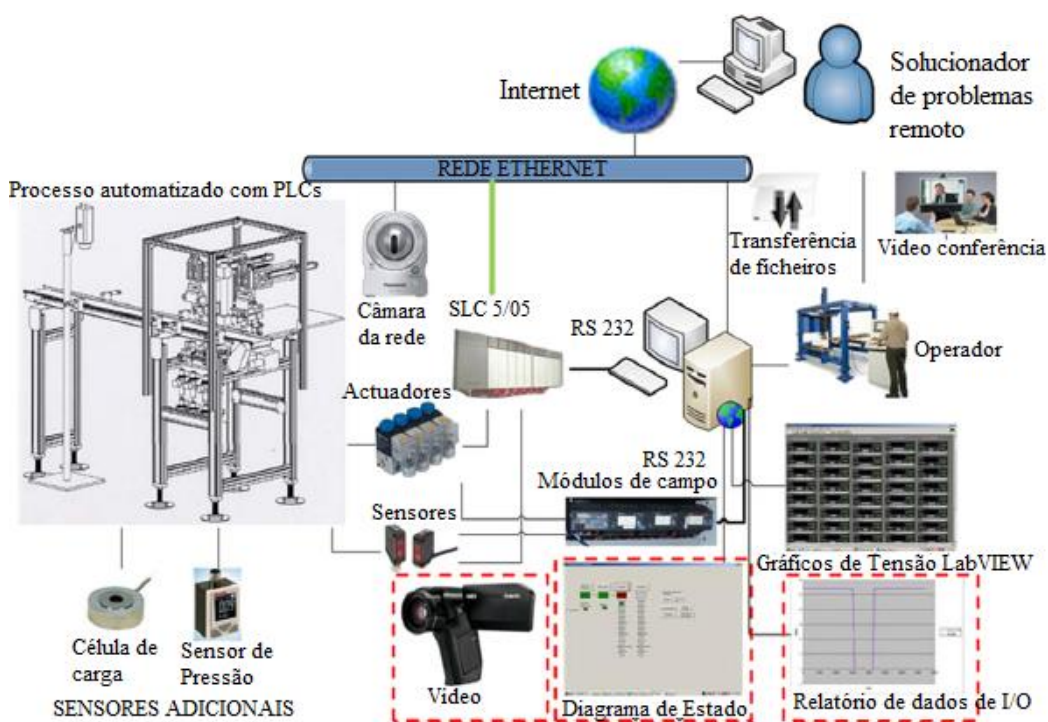


Figura 2.24: Arquitectura 3 do sistema de diagnóstico remoto de um processo controlado com PLCs da Universidade do Texas A&M, EUA; adaptado de [23].

O computador do cliente é equipado com o programa RSLogix® 500 para programar e monitorizar (o estado de I/Os), a partir da comunicação via Ethernet com o PLC. O cliente acede à rede Ethernet através de uma ligação VPN. A interface gráfica em LabVIEW é utilizada para monitorizar em tempo-real as tensões medidas em todas as entradas e saídas do processo automatizado. Esta interface gráfica pode ser acedida pelo cliente remoto [23].

Exemplo de Sistemas FDI em Portugal

Além destes exemplos, destacam-se em Portugal, igualmente na área de FDI, o trabalho de investigação desenvolvido por Palma [56] e o sistema proposto por Mendes [57].

O trabalho de Palma [56] está assente no desenvolvimento de métodos de detecção e diagnóstico de falhas que operam em tempo-real para sistemas controlados em anel fechado e com restrições temporais. As metodologias de detecção e diagnóstico de falhas propostas foram testadas em modelos de simulação e posteriormente em processos reais. Os algoritmos para a criação de modelos, simulação, controlo, detecção e diagnóstico de falhas, e supervisão foram implementados em ambiente Matlab®. Todas as abordagens propostas incluem um filtro de passa-baixo para obter a detecção de falhas e o isolamento de falhas dos sinais [56].

No sistema proposto em [57] são utilizadas técnicas de Inteligência Artificial Distribuída na construção de sistemas de controlo tolerante a falhas, através de uma arquitectura baseada em sistemas multi-agente. A aplicabilidade do sistema em processos industriais foi considerada com a proposta de uma nova plataforma e de uma nova ferramenta de desenvolvimento para ambiente Matlab/Simulink®. O sistema desenvolvido foi validado e testado em simulação e no dispositivo experimental do processo dos três tanques [57].

3. Sistema de Detecção e Diagnóstico de Falhas via Internet Proposto

Neste capítulo é feita, na secção 3.1., uma descrição dos programas e linguagens de programação utilizados para a implementação do sistema de detecção e diagnóstico de falhas via Internet do presente Trabalho Final de Mestrado. De seguida, na secção 3.2. é caracterizado o dispositivo utilizado para aquisição de dados do sistema proposto. Por fim, na secção 3.3. é dada uma breve explicação da arquitectura do sistema FDI proposto, em que a estrutura do sistema é do tipo Cliente/Servidor.

3.1. Programas e Linguagens de Programação Utilizados

Para o desenvolvimento da programação do sistema de detecção e diagnóstico proposto, foi escolhida a plataforma NI LabVIEW, devido à facilidade na integração do *hardware*, aquisição e processamento de sinais, e ainda os vários recursos de programação que resultaram na redução do tempo de desenvolvimento em comparação com outras plataformas.

Num laboratório científico ou educacional, bem como em processos industriais, a selecção de equipamentos é inevitavelmente heterogénea: I/Os digitais, motores, câmaras digitais e, muitas vezes, a mistura de tecnologias novas e outras relativamente antigas para conectar equipamentos aos computadores que gerem os sistemas. Geralmente, os sistemas de controlo e aquisição de dados estão frequentemente longe de ser desenvolvidos uma vez para sempre. Pelo contrário, são actualizados continuamente através da substituição e/ou introdução de novos componentes de forma a acompanhar a evolução da tecnologia e indústria. A programação gráfica do LabVIEW tem a capacidade de trazer a flexibilidade e modularidade através do seu elevado número de *drivers* e bibliotecas para análise de dados e exibição gráfica, permitindo facilmente a ampliação do sistema, no futuro, a outros processos industriais caso seja necessário.

Por outro lado, a utilização do LabVIEW permitiu uma outra abordagem diferente em relação às linguagens de programação existente no LabAPI – ISEL (JavaScript, PHP e HTML). Como também uma diversidade em relação às abordagens

seguidas em outros trabalhos de Mestrado até à data, o que possibilita comparação e alternativas científicas na abordagem prosseguida.

Por último, o LabVIEW é parte integrante do programa de Mestrado em Engenharia Mecânica, traduzindo-se assim numa oportunidade para aplicar os conhecimentos estudados e as competências desenvolvidas.

3.1.1. National Instruments LabVIEW

LabVIEW é uma linguagem de programação gráfica especificamente concebida para o desenvolvimento de sistemas de instrumentação, diagnóstico, e de aquisição de dados, sendo talvez o ambiente mais abrangente para estabelecer este tipo de sistemas [58]. Este tipo de linguagem de programação providencia soluções prontas a utilizar tanto para aquisição de dados como para controlo, para um grande número de equipamentos e para a análise de diversos tipos de dados. Os programas do LabVIEW são denominados instrumentos virtuais, ou VIs, por imitarem a aparência e modo de operação de instrumentos físicos, como osciloscópios e multímetros. Publicar um VI na Internet através de “clique-e-partilha” é há muito tempo uma característica deste programa [59].

Um ficheiro LabVIEW é denominado Projecto, é aqui que se aglomeram todos os VIs, controlos, bibliotecas de etiquetas, variáveis, entre outros, fundamentais ao funcionamento do programa desenvolvido. Na janela Project Explorer é possível adicionar, importar, exportar e eliminar, novos elementos no/do Projecto.

Um instrumento virtual é sempre constituído por duas janelas: painel frontal e diagrama de blocos. O painel frontal é uma janela de interface com o utilizador, ou seja, onde se visualizam os controladores e os indicadores gráficos, objectos, utilizados para cumprir o objectivo do programa desenvolvido.

O diagrama de blocos é o ambiente de computação, isto é, onde se coloca o código através da utilização de representações gráficas de funções para controlar os objectos do painel frontal. Entre os objectos do diagrama de blocos estão terminais, sub-VIs, funções, constantes, estruturas e fios, que transferem dados entre outros objectos do diagrama de blocos. Os objectos da janela do painel frontal são exibidos, no diagrama de blocos, como terminais. Terminais são portas de entrada e saída, que trocam informações entre o painel frontal e o diagrama de blocos.

Assim, é no diagrama de blocos que é feita a ligação entre controladores e indicadores (ver Figura 3.1).

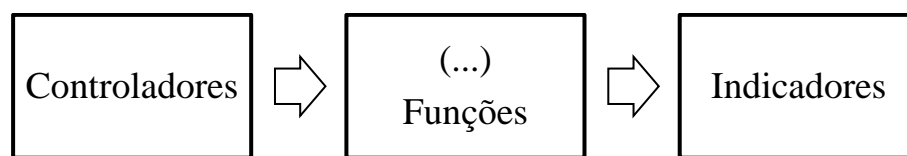


Figura 3.1: Fluxograma da relação entre controladores e indicadores no diagrama de blocos

Assim como outras linguagens de programação, o programa LabVIEW trabalha com diferentes tipos de dados e diversas maneiras de organizá-los. Dados numéricos, strings e booleanos são alguns dos exemplos dos diferentes tipos de dados utilizados no LabVIEW. Esses tipos de dados podem ser identificados pela cor e *layout*, e são usados com objetivos específicos [60].

3.1.2. OPC Servers

O Object Linking and Embedding (OLE) para Processos de Controlo (OPC) define o padrão para a comunicação de dados de processos industriais em tempo-real entre dispositivos de controlo e interfaces humano-máquina (HMIs). O National Instruments OPC Servers é um programa que converte protocolos industriais proprietários para os protocolos abertos OPC Classic e OPC Unified Architecture (UA). Essa conversão permite a interface do programa LabVIEW com dispositivos de automação industrial, como controladores lógicos programáveis e dispositivos de outros fornecedores (por exemplo, controladores de automação programável – Programmable Automation Controllers – PACs) [61]. A National Instruments providencia uma solução de servidor OPC, com o OPC Servers, que contém uma lista de drivers para muitos dos PLCs na indústria, nomeadamente os controladores lógicos programáveis Siemens S7-1200 utilizados no presente Trabalho Final de Mestrado.

A Figura 3.2 exhibe a interacção existente entre o PLC, o OPC Server e o LabVIEW. Deste modo, é possível perceber que um programa OPC Server usa a *driver* proprietária do controlador lógico programável para criar etiquetas OPC (tag OPC) para cada Entrada/Saída no PLC que por sua vez, se ligam às variáveis partilhadas (*Shared Variables*). O OPC possibilita que as aplicações do servidor e do cliente partilhem informação entre si em tempo-real (por exemplo: o estado real de funcionamento dos processos) através dos servidores de I/Os OPC Client,

incluídos no módulo LabVIEW Datalogging and Supervisory Control (DSC). Os servidores de I/Os OPC Client são configurados no SVE que providencia um endereço para cada etiqueta OPC, dando origem a uma variável partilhada. A partir do momento que as variáveis partilhadas recebem os valores, o LabVIEW consegue facilmente ler e escrever as/nas variáveis partilhadas criadas [61].

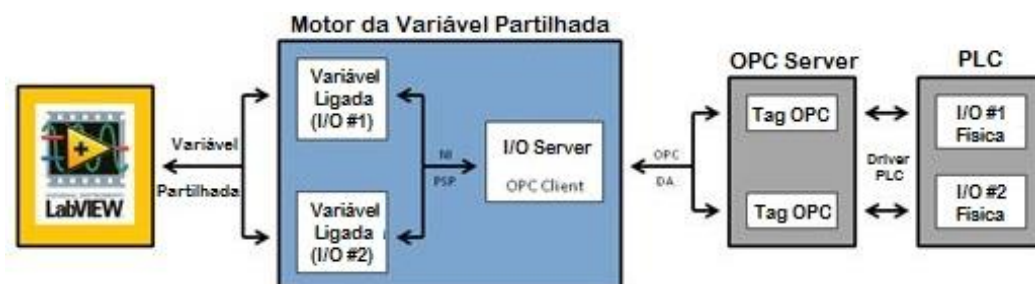


Figura 3.2: LabVIEW e a rede OPC com o módulo DSC, adaptado de [61].

O DSC é um módulo do programa LabVIEW que tem como característica fundamental a capacidade de configurar dados de supervisão, de controlo e de aquisição de dados e de sistemas com uma grande quantidade de I/Os. É através deste módulo que é possível criar as variáveis partilhadas no programa desenvolvido em LabVIEW. Outra das vantagens da utilização do DSC é a possibilidade de registar dados adquiridos (relatórios, históricos, entre outros) numa base de dados. Além disso, com este módulo é viável configurar o registo de alarmes e notificações sem programação adicional [62].

3.2. Aquisição de Dados – NI USB-6008

A parte física do dispositivo de aquisição de dados (Data Acquisition – DAQ) é a interface entre o computador e o mundo exterior. Essencialmente, funciona como um equipamento que digitaliza sinais analógicos de entrada para que o computador os possa interpretar. No sistema de diagnóstico de falhas proposto é utilizada a placa portátil de aquisição de dados NI USB-6008 da National Instruments, representada na Figura 3.3. A conexão entre o dispositivo de DAQ e o computador onde o programa corre é feita através de uma porta USB, isto é, sem a necessidade de uma fonte de alimentação extra. É instalada uma *driver* de comunicação entre a placa DAQ NI USB-6008 e o programa em LabVIEW (programa da aplicação) – NI-DAQmx – que previne que se tenha que recorrer a programação de baixo nível e a programação de comandos complicados de forma a ter acesso às funções do *hardware*. Outra das vantagens da utilização da placa DAQ NI USB-6008 é o

facto do programa assistente de aquisição de dados, incluído com a *driver* NI-DAQmx, ter uma interface gráfica e interactiva com o programador para facilmente configurar, testar e adquirir dados medidos. Por outro lado, a *driver* NI-DAQmx pode ser instalada em vários sistemas operativos incluindo Windows, Linux e Mac OS X.



Figura 3.3: Placa DAQ NI USB-6008 do sistema proposto.

A NI USB-6008 tem as seguintes especificações:

- 8 entradas analógicas (12-bit, 10 kS/s);
- 2 saídas analógicas (12-bit, 150 S/s);
- 12 I/Os digitais;
- Compatível com LabVIEW, LabWindows/CVI, e Measurement Studio para Visual Studio NET.

A Figura 3.4 representa a disposição dos pinos da placa de aquisição de dados NI USB-6008. Os nomes dos sinais de entrada analógicos estão classificados como nomes de entrada analógica simples (por exemplo AI 0 e AI 4), e depois de nomes de entrada analógica diferencial (por exemplo AI 0+/-). A Tabela 3.1 contém uma descrição detalhada de cada pino/sinal.

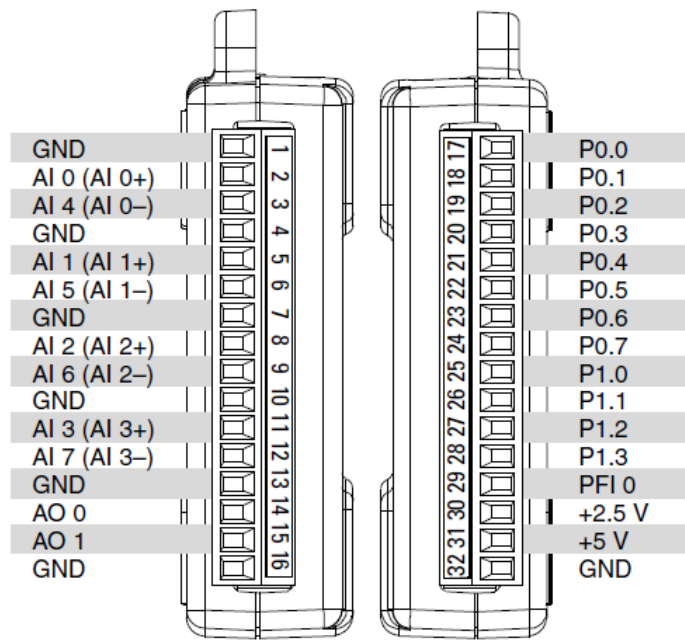


Figura 3.4: Diagrama da disposição dos pinos da placa DAQ NI USB-6008 [63].

Tabela 3.1: Descrição dos sinais da placa DAQ NI USB-6008; adaptado de [63].

Nome do sinal	Referência	Direcção	Descrição
GND	–	–	Terra.
AI <0.. 7>	Varia	Entrada	Canais de 0 a 7 de entrada analógica. Para medições simples, cada sinal é um canal de tensão de entrada analógica. Para medições diferenciais AI 0 e AI 4 são as entradas positiva e negativa do canal 0 de entrada analógica diferencial. Os seguintes pares de sinais também formam canais de entrada diferencial: AI <1, 5>, AI <2, 6> e AI <3, 7>.
AO <0, 1>	GND	Saída	Canais 0 e 1 de saída analógica. Suprime a tensão de saída analógica do canal 0 ou do canal 1.
P0.<0.. 7>	GND	Entrada/Saída	Canais 0 a 7 da porta digital I/O 0. Cada sinal pode ser configurado como entrada ou saída.
P1.<0.. 3>	GND	Entrada/Saída	Canais 0 a 3 da porta digital I/O 1. Cada sinal pode ser configurado como entrada ou saída.
PFI 0	GND	Entrada	Este pino é configurável como canal digital ou como entrada de contador de eventos.
+2.5 V	GND	Saída	Referência externa.
+5 V	GND	Saída	Fonte de alimentação. Providencia uma alimentação de +5 V até 200 mA.

3.3. Arquitectura do Sistema Proposto

A arquitectura da rede proposta para o sistema de detecção e diagnóstico de falhas com aplicabilidade em processos industriais e testado no LabAPI – ISEL está ilustrada na Figura 3.5. A estrutura do sistema é do tipo Cliente/Servidor e consiste em cinco elementos fundamentais:

- Um computador servidor (10.68.13.171) que aloja o *website* do LabAPI – ISEL (<http://10.68.13.171/isel/menu.php>), e a base de dados onde é gerido o acesso dos utilizadores às páginas web da experiência cilindros e da experiência pórtico (a gestão de acessos à página do sistema FDI é gerido pelo próprio sistema);
- Três PLCs Siemens S7-1200;
- Um computador que detém a programação em LabVIEW do sistema FDI, a programação dos PLCs das experiências (Tia Portal), o OPC Servers que permite a interface do LabVIEW com os S7-1200, e hospeda o *website* do sistema de FDI;
- Um transdutor de pressão;
- E uma placa de aquisição de dados NI USB-6008 conectada ao computador de ambiente remoto e que adquire os dados recolhidos do transdutor de pressão;

Além do descrito anteriormente, o servidor (10.68.13.171) acomoda a *webcam* e o computador também serve como ambiente de trabalho remoto.

A segurança da comunicação de dados da arquitectura proposta é garantida por uma *Firewall* e por uma ligação VPN. A rede pode ser acedida por um cliente interno (dentro da rede do ISEL), acedendo directamente ao servidor, ou por um cliente externo (fora da rede do ISEL). Para o cliente externo aceder ao LabAPI – ISEL, e conseqüentemente ao sistema FDI proposto, terá que o fazer através da ligação VPN do ISEL (<http://www.isel.pt/pInst/eISEL/Portal/AUR.html>). Após a autorização de acesso, o cliente externo poderá aceder ao servidor como um cliente interno.

As estações do cliente são do tipo computadores localizados na zona de trabalho dos utilizadores (que seja interno ou externo). A comunicação entre o servidor e as estações do cliente é suportado por rede de área local (Ethernet).

É a partir do *website* do LabAPI – ISEL que os utilizadores têm acesso à página do sistema de detecção e diagnóstico de falhas proposto no presente trabalho, bem como às restantes páginas previamente desenvolvidas em [8] e [9]. A estrutura do *website* está representada na Figura 3.6, onde também se indica a localização das páginas *web* nos respectivos elementos físicos. A página do sistema FDI proposto (<http://10.68.11.55:8082/FDI.html>), bem como as restantes páginas representadas na Figura 3.6 podem ser visualizadas no Anexo A, página 123.

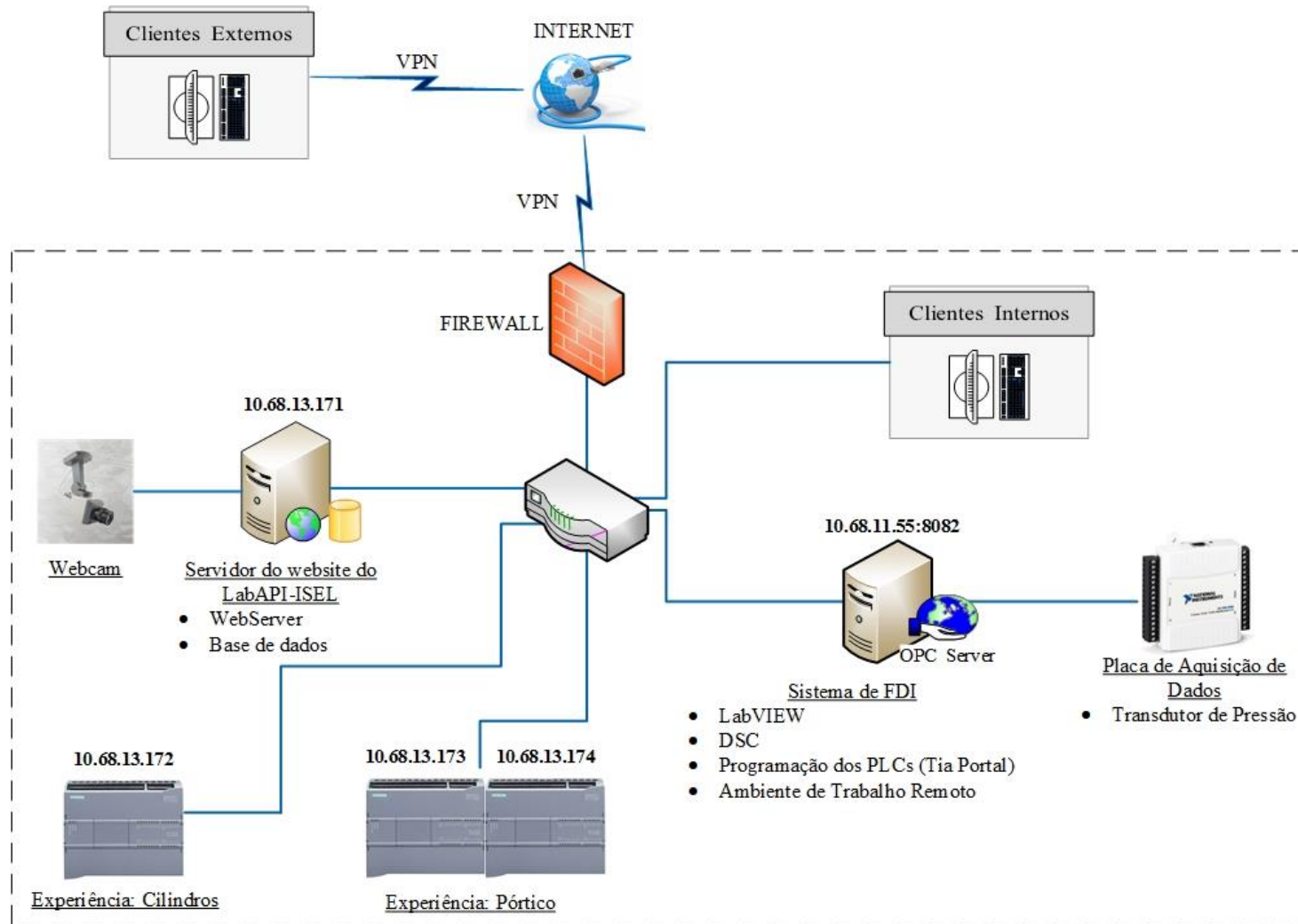


Figura 3.5: Arquitectura da rede do sistema de detecção e diagnóstico de falhas proposto.

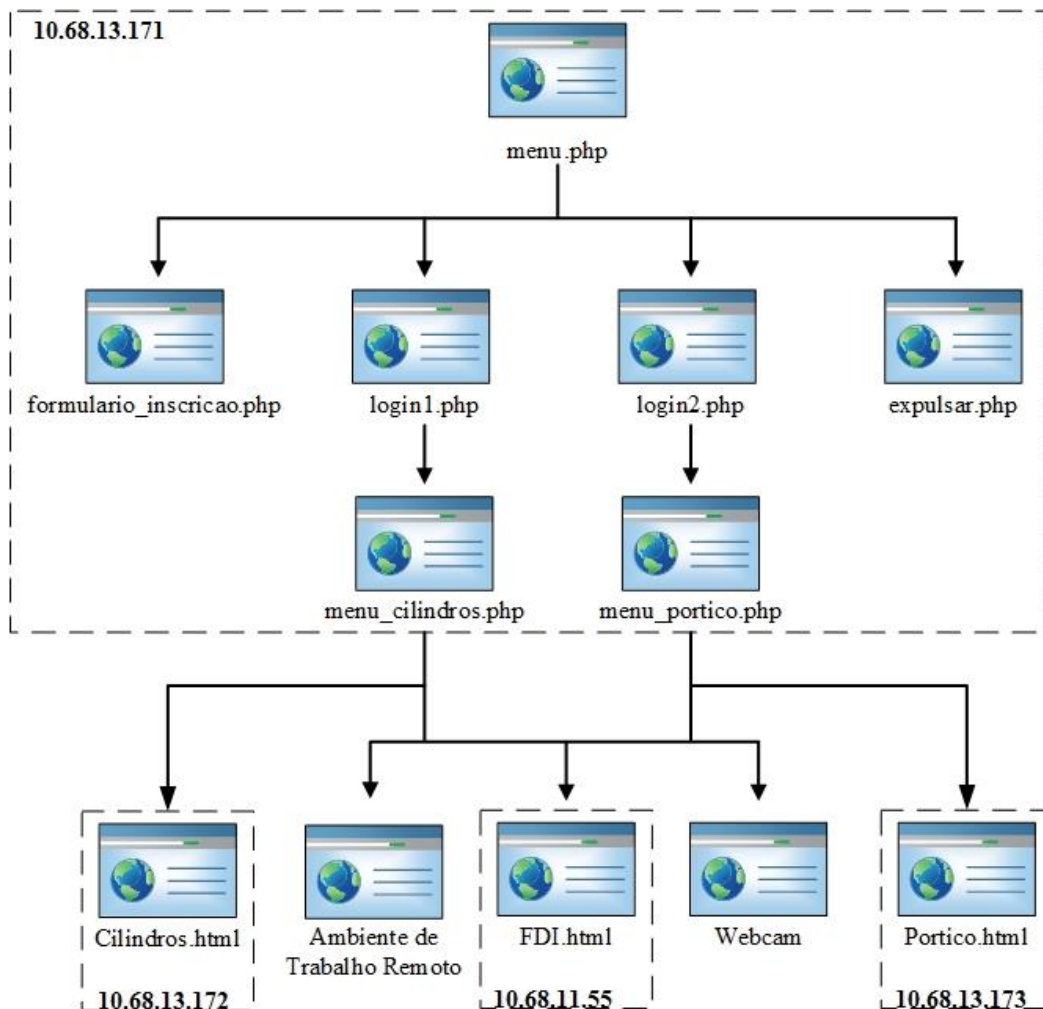


Figura 3.6: Estrutura do *website* do LabAPI – ISEL.

A página de Internet do sistema de detecção e diagnóstico de falhas proposto foi publicada através da *Web Publishing Tool* (ferramenta de publicação na *web*) do LabVIEW, que providencia o controlo de toda a aplicação através da internet. A *Web Publishing Tool* tem várias características e uma delas é o controlo remoto do painel frontal utilizado neste sistema. O LabVIEW neste caso actua como um servidor e proporciona aos utilizadores da Internet (internos ou externos) um painel frontal onde podem controlar facilmente o sistema FDI como se estivessem no computador onde os instrumentos virtuais residem e executam. Quando o utilizador faz o login na página *web*, o *browser* da Internet vai automaticamente correr o painel frontal remoto do sistema FDI da mesma forma que uma aplicação Java o iria correr. Assim, o LabVIEW necessita de um motor tal como uma aplicação necessita de um motor Java. É aqui que residem algumas limitações deste sistema.

O cliente externo ou interno não necessita de ter o programa LabVIEW instalado para controlar o painel frontal remoto, no entanto é indispensável que este instale o LabVIEW Run-Time Engine, caso contrário apenas irá visualizar ou monitorizar uma imagem instantânea que é actualizada constantemente. Quando o cliente não possui este módulo instalado, este é automaticamente redireccionado para o *website* da National Instruments para fazer o download do módulo necessário.

Outra das limitações é o facto do painel frontal remoto utilizar *plugins* NPAPI que não são suportados pelo *web browser* Google Chrome versão 42 ou superior, ou pelo Microsoft Edge. Deste modo, apenas é possível utilizar os seguintes *web browsers* que constituem 27,56% da utilização mundial, segundo dados de Agosto de 2016 [64]:

- Internet Explorer 5.5 ou superior;
- Netscape 4.7 ou superior;
- Mozilla Firefox;
- Safari;
- Opera.

Por último, os elementos da janela são possíveis de visualizar na sua totalidade recorrendo à *scrollbar* ou reduzindo o tamanho da página no *web browser*.

3.3.1. Arquitectura do Programa

Considerando os recursos existentes no laboratório de Automação Industrial presente na Área Departamental de Engenharia Mecânica, ISEL, e que as variáveis a monitorizar nos processos industriais são, normalmente, na sua maioria lógicas e na minoria analógicas, definiu-se o desenvolvimento de um sistema de detecção e diagnóstico de falhas via Internet com aplicabilidade em processos industriais, em que a detecção está assente no método de verificação de limites fixos e o diagnóstico realizado a partir de árvores de falha de lógica booleana e regras IF-THEN.

Posto isto, iniciou-se um estudo dos processos e variáveis presentes nas experiências “Pórtico” e “Cilindros” (processos industriais existentes no LabAPI – ISEL) a fim de verificar quais as falhas que poderiam ocorrer em cada experiência e dessas quais seriam possíveis monitorizar, detectar e diagnosticar.

Findo este processo, foi possível chegar às falhas identificadas, descritas e classificadas na Tabela 3.2 para a experiência “Pórtico” e na Tabela 3.3 para a experiência “Cilindros”.

Tabela 3.2: Falhas e natureza de falhas da experiência “Pórtico”.

Falha	Natureza de Falha	Classificação
Electroíman não liga.	Avaria do sensor de descida, isto é, sensor de descida a 0 indevidamente.	Falha de sensor.
	Avaria no electroíman.	Falha no actuador.
	O pórtico não está na posição “descida” (sensor de descida a 0).	-
	Fio de ligação dos actuadores desconectados das entradas/saídas dos PLCs.	Falha de processo.
Pórtico não chega à posição pretendida pelo utilizador.	Avaria no(s) motor(s) impossibilitando que o pórtico chegue ao destino escolhido pelo utilizador.	Falha de actuador.
	Avaria no sensor da posição de chegada, estando este a 1, inviabilizando o <i>set</i> e respectivo <i>output</i> da variável do movimento pretendido.	Falha de sensor.
	Fio de ligação dos actuadores desconectados das entradas/saídas dos PLCs.	Falha de processo.

Tabela 3.3: Falhas e natureza de falhas da experiência “Cilindros”.

Falha	Natureza de Falha	Classificação
Cilindro não avança/recua.	Fuga na rede pneumática. Compressor desligado.	Falha de processo.
	O modo manual não está ligado.	-
	Avaria do cilindro.	Falha de actuadores.
	Avaria da válvula de solenoide.	
Ciclo 1 não inicia.	O modo automático não está ligado.	-
	Avaria no(s) cilindro(s).	Falha de actuadores.
	Avaria na(s) válvula(s) solenoide.	

A verificação de limites fixos é realizada através da monitorização das variáveis lógicas de entrada e saída dos processos industriais testados (experiências “Pórtico” e “Cilindros”) no LabAPI – ISEL, e de monitorização da pressão da rede pneumática da experiência “Cilindros”.

Adicionalmente, previu-se a implementação de um sistema de monitorização e análise de vibrações, com a aquisição ou simulação de um sinal medido. Ao sinal adquirido são aplicados métodos de detecção de falhas por modelos de sinal, onde o mesmo é analisado no domínio do tempo e no domínio da frequência. A análise no domínio do tempo é feita com recurso ao cálculo de características a partir do sinal no tempo (RMS, factor de crista e nível global), e em relação ao domínio da frequência é realizada uma análise do espectro FFT.

A Figura 3.7 mostra a inserção e a divisão do sistema FDI proposto nos processos das experiências existentes no LabAPI – ISEL, incluindo o módulo de autenticação do utilizador/cliente.

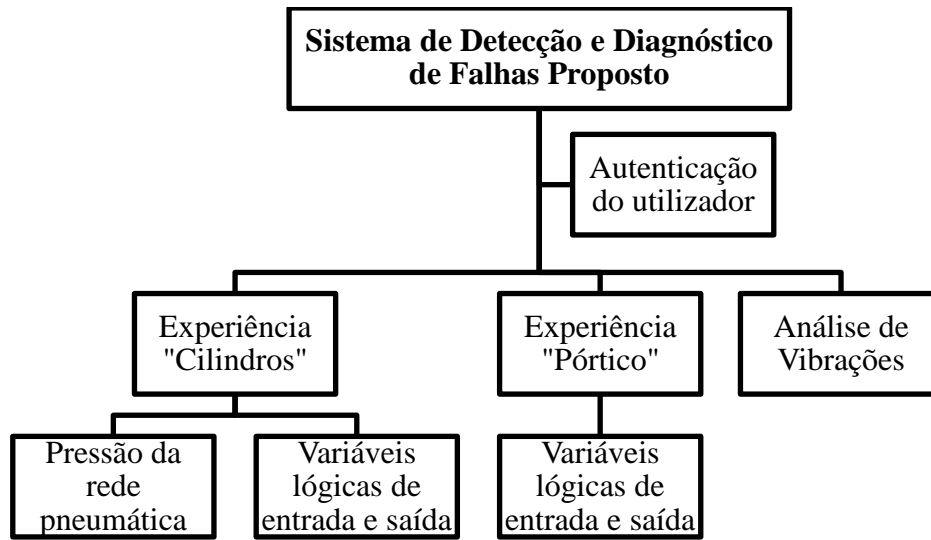


Figura 3.7: Organograma da arquitectura do sistema de detecção e diagnóstico de falhas proposto testado no LabAPI – ISEL.

4. Resultados Experimentais

O presente capítulo contextualiza o Trabalho Final de Mestrado em termos da sua implementação e inserção no *website* do LabAPI – ISEL, bem como é apresentada e descrita a metodologia e o desenvolvimento do sistema de FDI via Internet proposto.

4.1. Configuração das Etiquetas OPC

A interface entre o LabVIEW e os PLCs Siemens S7-1200 utilizados para controlar as experiências “Cilindros” e “Pórtico”, é realizada através do programa OPC Server descrito na secção 3.1.2. OPC Servers (página 49), das etiquetas OPC criadas pelo mesmo e das variáveis partilhadas. Primeiramente, foi configurado o canal, ao qual foi dado o nome “Siemens” e onde se estabelece a *driver* de comunicação dos equipamentos que é do tipo TCP/IP Ethernet. De seguida, foram caracterizados os PLCs definindo nome, modelo e endereço IP de cada um. A Tabela 4.1 contém as propriedades estabelecidas na configuração dos autómatos programáveis no OPC Server.

Tabela 4.1: Propriedades dos PLCs Siemens S7-1200 das experiências “Cilindros” e “Pórtico”.

Experiência	Nome	Modelo	Endereço IP
“Cilindros”	172	S7-1200	10.68.13.172
“Pórtico”	173		10.68.13.173
	174		10.68.13.174

Para cada PLC e respectivas variáveis físicas foram criadas as etiquetas OPC correspondentes, inserindo os seguintes atributos:

- ✓ nome, por exemplo “input_0”;
- ✓ endereço, por exemplo, “I0.0”;
- ✓ o tipo de dados, que para as experiências testadas são na sua maioria variáveis booleanas;
- ✓ e o tipo de acesso, que para todas as variáveis é do tipo ler/escrever.

A Figura 4.1 permite visualizar o painel de configuração das etiquetas OPC e de como estas são organizadas e inseridas no canal (Siemens) e nos equipamentos respectivos (172, 173 e 174), e necessários para a monitorização e controlo das experiências testadas.

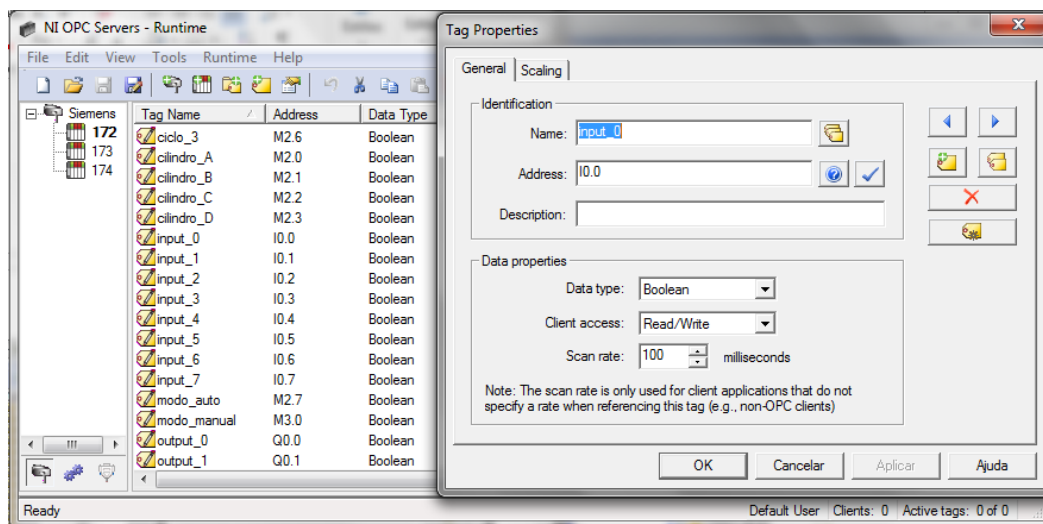


Figura 4.1: Configuração das etiquetas OPC.

Após a configuração das etiquetas OPC, é possível monitorizar em tempo-real os dados das mesmas através do Quick Client, presente no NI OPC Servers, isto é, é possível consultar rapidamente o valor da variável booleana (0 ou 1) (ver Figura 4.2). O NI OPC Servers e o Quick Client pode ser acedido pelos utilizadores do sistema de diagnóstico de falhas proposto, a partir do ambiente remoto do *website* do LabAPI – ISEL.

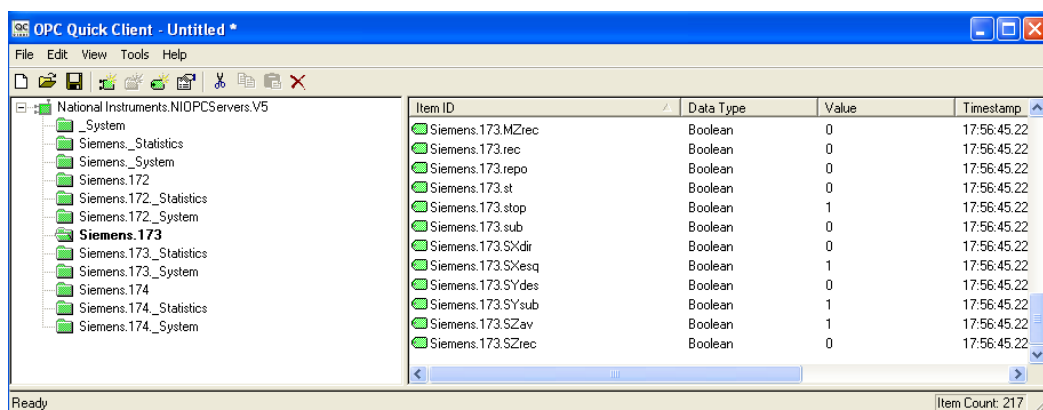


Figura 4.2: Janela de interface OPC Quick Client.

Configuração das Variáveis Partilhadas

Na janela de Projecto do programa LabVIEW foi adicionado um servidor de I/Os OPC Client que consequentemente criou, de forma automática, uma biblioteca para variáveis partilhadas utilizada para gerir o servidor de I/Os. As variáveis partilhadas são usadas para a conexão com as etiquetas OPC dos autómatos programáveis, sendo associadas antes si através da janela *Create Bound Variables* do programa LabVIEW. Assim, as variáveis partilhadas ficam disponíveis na rede para outros computadores e clientes OPC, permitindo o acesso aos dados dos PLCs das experiências nativamente no LabVIEW, estando prontas a ser adicionadas aos VIs necessários para o desenvolvimento do sistema FDI. A Figura 4.3 representa a biblioteca das variáveis partilhadas dos PLCs 172, 173 e 174 utilizados para controlar as experiências “Cilindros” e “Pórtico”, bem como ilustra a representação gráfica de uma variável partilhada do PLC “172” no diagrama de blocos do LabVIEW.

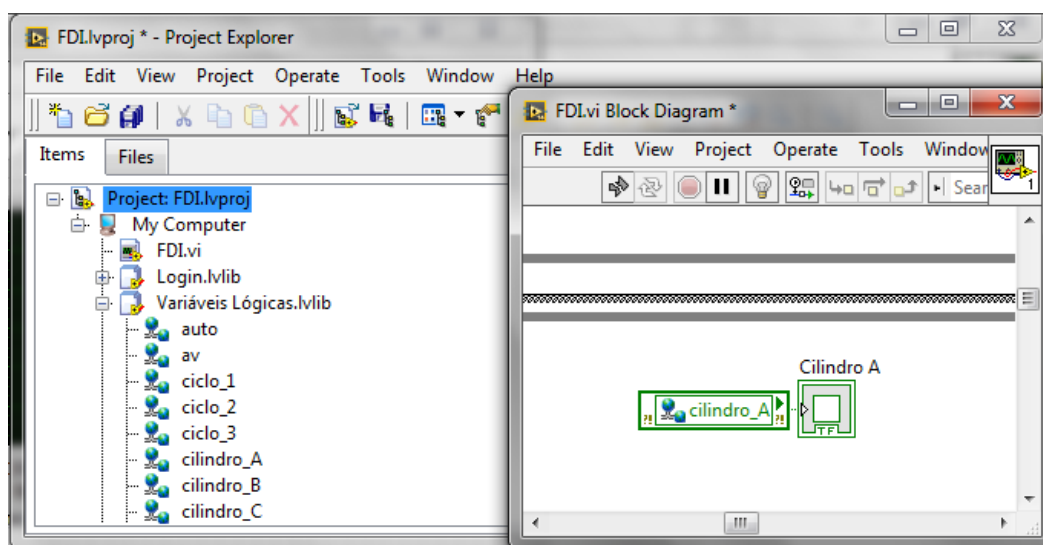


Figura 4.3: Biblioteca de variáveis partilhadas e etiqueta de variável partilhada do PLC “172” no diagrama de blocos.

4.2. Autenticação do Utilizador

O ficheiro do projecto LabVIEW do sistema de detecção e diagnóstico de falhas proposto aglomera o VI principal “FDI”, os sub-VIs dele dependentes, as bibliotecas das variáveis lógicas, entre outros fundamentais ao funcionamento do programa desenvolvido. Na Figura 4.4 visualiza-se a janela do explorador do projecto do sistema proposto.

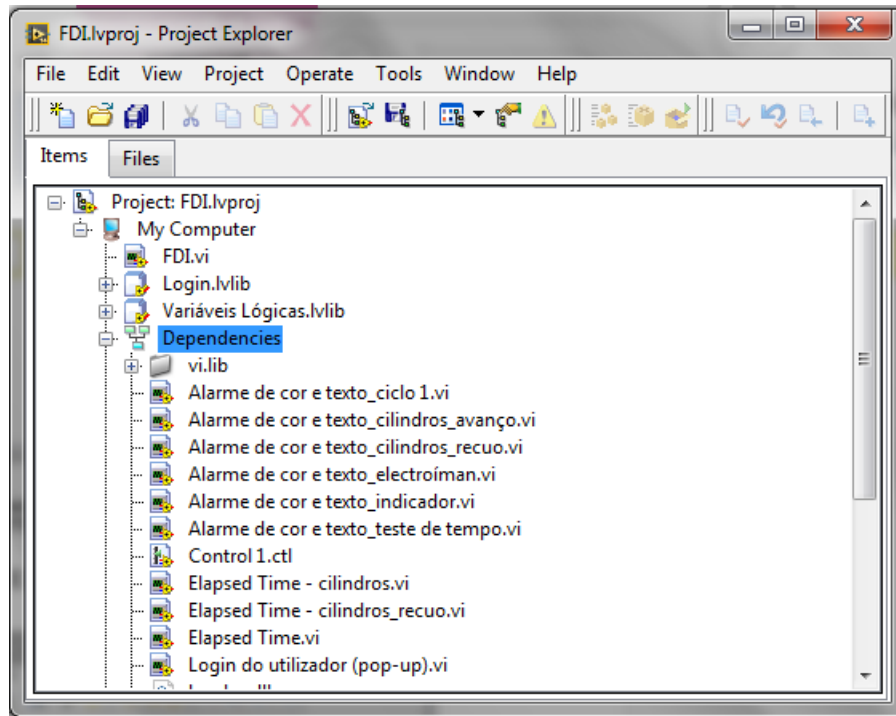


Figura 4.4: VI principal “FDI” e os sub-VIs dependentes no projecto LabVIEW do sistema FDI proposto.

O acesso por parte do utilizador às diferentes aplicações Experiência “Pórtico”, Experiência “Cilindros” e Análise de Vibrações é realizado a partir de *Tab Controls* (controlos com abas), que o utilizador pode percorrer e seleccionar consoante o que pretende monitorizar. No entanto, o utilizador necessita de fazer o *login* para aceder às diferentes propriedades do painel frontal do sistema FDI proposto.

Para a autenticação dos utilizadores foi definida a distinção entre utilizador do tipo “Aluno” e utilizador do tipo “Admin”, com palavras-passe distintas. O utilizador do tipo “Admin” tem privilégios de administrador e por isso tem acesso a todas as funcionalidades do sistema de monitorização, detecção e diagnóstico de falhas, nomeadamente a interrupção da execução do sistema e a interrupção das experiências em caso de emergência. Pelo contrário, o “Aluno” apenas pode monitorizar e modificar os estados dos processos e das suas variáveis lógicas de entrada e saída, e os parâmetros das harmónicas e o tipo de aquisição de dados, no caso do sistema de análise de vibrações. Este processo de autenticação de utilizador é executado com a criação de um sub-VI “Login do utilizador (pop-up)”. A Figura 4.5 representa o diagrama de blocos do sub-VI “Login do utilizador (pop-up)”.

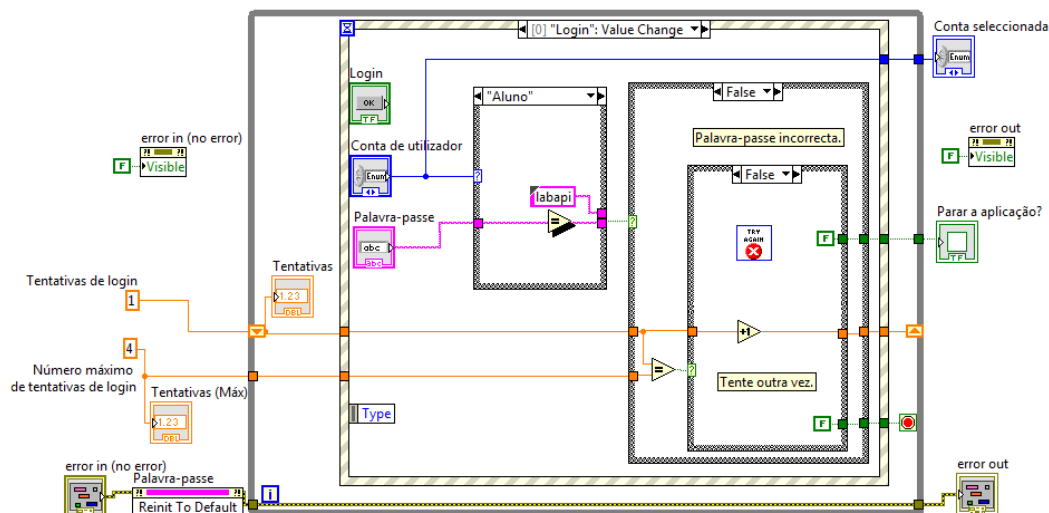


Figura 4.5: Diagrama de blocos do sub-VI de autenticação do utilizador.

A possibilidade de escolha do tipo de conta por parte do utilizador é conseguida com recurso a uma estrutura do tipo *Case*. As estruturas *Case* do LabVIEW permitem executar programaticamente secções de código gráfico com base em determinadas condições. Uma estrutura *Case* pode ter dois ou mais subdiagramas, ou “casos”. Cada subdiagrama é visualizado individualmente quer no diagrama de blocos, quer no painel frontal, bem como é executado individualmente. Esta estrutura é similar aos comandos “if, then, else, ...” de outras linguagens de programação. A *Case* é então controlada através do terminal selector, colocado no exterior da mesma, que determina o subdiagrama a ser executado mediante o valor de entrada. O terminal selector da estrutura prevista é por sua vez controlado através de um controlo do tipo enumerado (“Conta de Utilizador”), que no painel frontal se assemelha a um menu com uma lista de dois itens (“Admin” e “Aluno”), visível na Figura 4.6, permitindo ao utilizador percorrer e fazer a sua selecção. O controlo do tipo enumerado associa cada caso aos valores dos itens de controlo e é representada pela cor azul no diagrama de blocos (ver Figura 4.5).

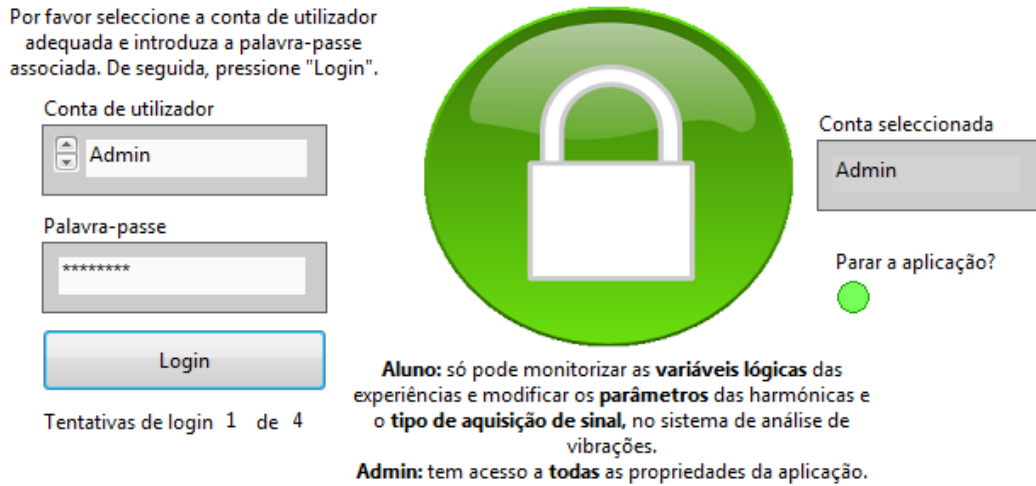


Figura 4.6: Painel frontal do sub-VI de autenticação do utilizador.

Deste modo, para a escolha entre dois tipos de conta por parte do utilizador, são caracterizados dois casos distintos para a estrutura *Case*. Em que, a palavra-passe para cada utilizador é definida no subdiagrama correspondente através de *strings*³, representadas pela cor rosa no diagrama de blocos. No painel frontal, os dados *string* são representados numa caixa de entrada do tipo *password*, ou seja, quando inseridos os caracteres, estes são representados por asteriscos.

O resultado da comparação entre a *string* definida em cada caso, no diagrama de blocos, e os caracteres inseridos na caixa de entrada, no painel frontal, define o valor de entrada do terminal selector da estrutura *Case* que permite/não permite a autenticação do utilizador e conseqüente acesso ao sistema FDI. Caso o valor de entrada seja verdadeiro (*True*), o que significa que a palavra-passe inserida está correcta, a janela *pop-up* de autenticação de utilizador encerra automaticamente permitindo ao utilizador aceder e explorar o sistema FDI consoante os seus privilégios. Por outro lado, caso o valor seja falso (*False*), isto é, caso a palavra-passe inserida pelo utilizador esteja incorrecta será exibida uma nova janela *pop-up* (o painel frontal e o diagrama de blocos deste sub-VI encontra-se no Anexo B, página 125) alertando o utilizador para o seu erro e incentivando a uma nova tentativa de autenticação. A janela de aviso de palavra-passe incorrecta encerra com um simples clique em cima da mesma, exibindo-se novamente a aplicação de autenticação do utilizador.

³ *String* é uma sequência de caracteres ASCII, que podem ser visíveis na janela ou não. *Strings* oferecem um formato independente de plataforma para informações e dados. No painel frontal, os dados *string* são exibidos como tabelas, caixas de entrada de texto e títulos [42].

O aluno ou administrador terá no máximo quatro tentativas erradas de *login*, sendo que à quarta e última tentativa será exibida uma janela *pop-up* (o painel frontal e o diagrama de blocos deste sub-VI encontra-se no Anexo B, página 125) alertando para esse facto e informando o encerramento da aplicação. Esta propriedade é alcançada através de um *While Loop* e da criação de um *Shift Register* associado ao mesmo. O *While Loop* é uma estrutura de execução de *loop* gráfico, que contém secções de código gráfico e permite que este seja executado repetidamente. O *Shift Register* permite transferir valores de uma iteração para outra e tem a aparência de um par de terminais em forma de seta, directamente opostos um ao outro nos lados verticais do *While Loop* (ver Figura 4.5). O terminal direito contém uma seta para cima que ao completar uma iteração armazena os dados da mesma. O LabVIEW transfere os dados armazenados no terminal direito para o terminal esquerdo. Assim, ao terminal esquerdo foi conectada uma constante numérica com início em 1, que simboliza o número de tentativas de *login* realizadas. A estrutura *Case* que determina se o número máximo de tentativas de autenticação de utilizador foi alcançado, tem como valor de entrada o resultado da função de comparação de igualdade entre o número de tentativas de *login* na actual iteração e o número máximo de tentativas de *login* (4). Quando a igualdade é falsa o subdiagrama correspondente incrementa em 1, através de uma função numérica de incremento conectada ao *Shift Register*, os dados da próxima iteração, isto é, o número de tentativas de *login* já realizadas. Quando a igualdade é verdadeira, ou seja, quando o número máximo de tentativas *login* é atingido, o subdiagrama correspondente executa e a janela *pop-up* de aviso de número máximo de tentativas de *login* atingido é exibida. Por fim, só é possível a utilização do sistema FDI por um cliente de cada vez, seja ele externo ou interno.

A inserção dos sub-VIs, mencionados anteriormente, no diagrama de blocos do VI principal “FDI” é processada através de uma estrutura *Event* que contém dois eventos: “*Login*” e “*Stop*” (ver Figura 4.7). O evento “*Stop*” encerra a aplicação de autenticação de utilizador caso o *login* seja efectuado com sucesso ou o número máximo de tentativas de *login* tenha sido atingido. Quando o evento “*Login*” ocorre, é exibido o diálogo de autenticação de utilizador e a estrutura executa o caso apropriado (“Admin” ou “Aluno”), que é determinado a partir do sub-VI “*Login do utilizador (pop-up)*” e da escolha feita por parte do cliente interno ou externo.

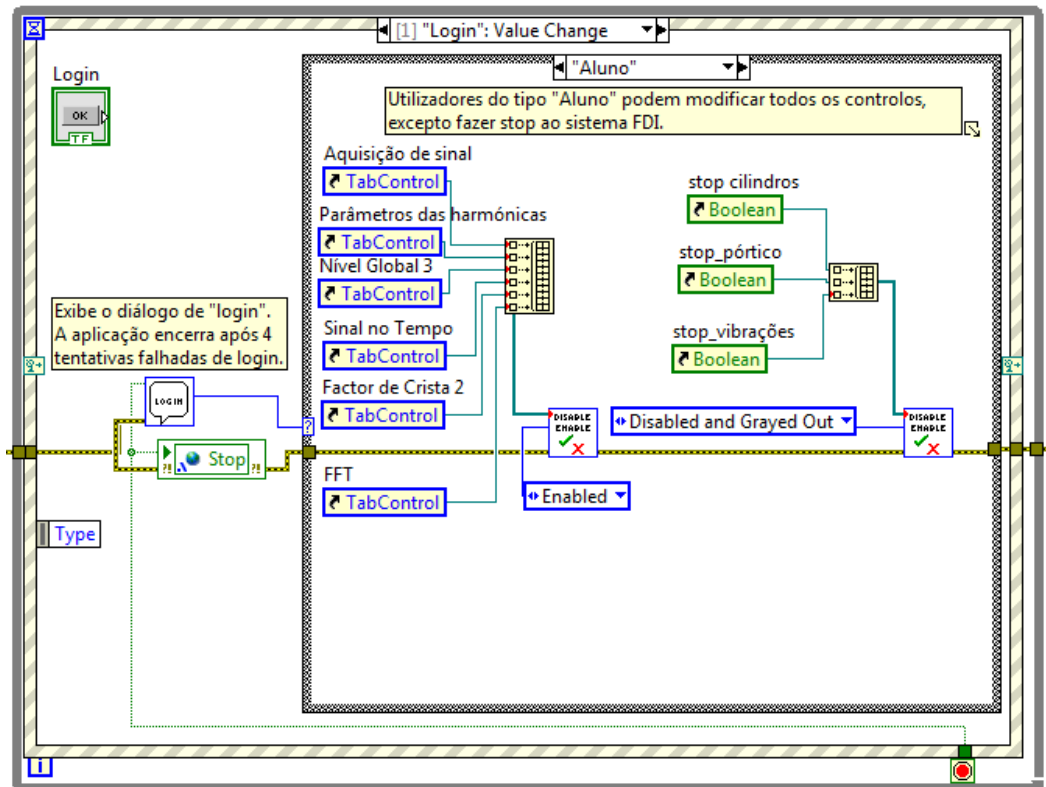


Figura 4.7: Chamada dos sub-VIs de autenticação do utilizador, e permitir e não permitir no diagrama de blocos do VI principal.

A estrutura *Case* determina as permissões por parte dos diferentes tipos de cliente (“Admin” ou “Aluno”), impossibilitando o acesso a algumas funcionalidades do sistema proposto, particularmente no caso do utilizador interno/externo do tipo “Aluno” que não pode parar as aplicações que correm no sistema FDI. Apenas a conta “Admin” tem permissões para fazer “stop” ao sistema FDI. A gestão de permissões é realizada com recurso ao sub-VI “Não permitir_Permitir” (ver Figura 4.8) Este sub-VI desactiva e desvanece quaisquer controlos cuja referência seja enviada para o mesmo. As referências dos controlos a activar/desactivar são inseridos no VI principal e conectados ao sub-VI “Não Permitir_Permitir” através de um *Array* de dados, que permite agregar múltiplos elementos.

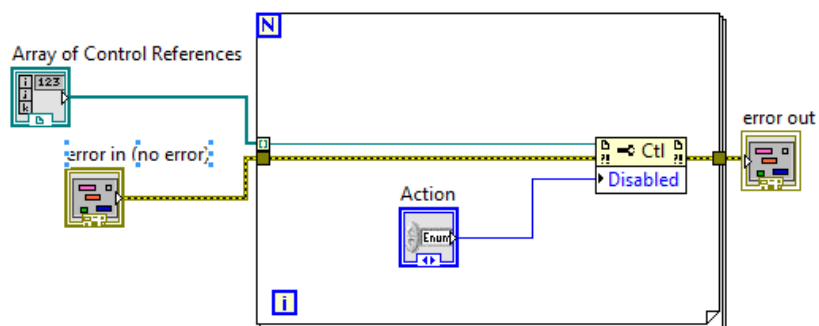


Figura 4.8: Sub-VI permitir e não permitir no diagrama de blocos.

4.3. Experiência “Pórtico”

A experiência “Pórtico” consiste no controlo do pórtico, representado na Figura 4.9, que é constituído por três motores que controlam a posição do electroíman num eixo tridimensional, o electroíman, seis sensores de fim de curso (dois para eixo) e 13 botões.

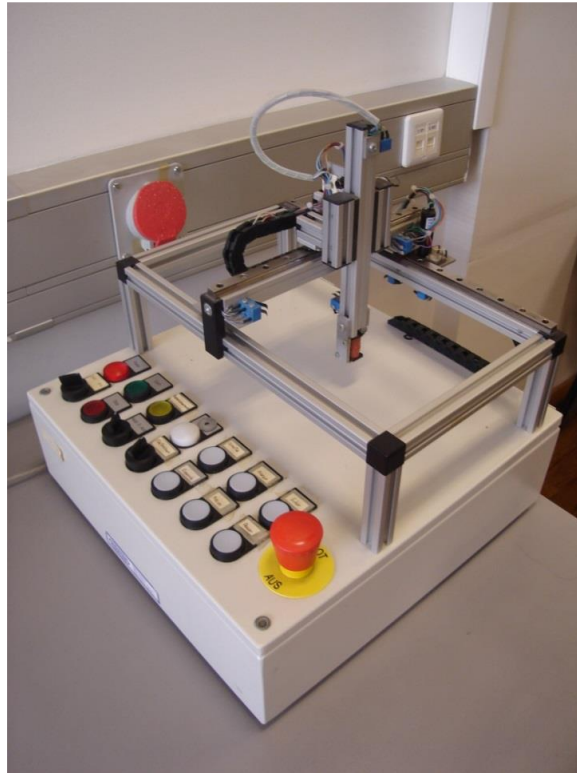


Figura 4.9: Experiência “Pórtico”.

O sistema FDI proposto aplicado à experiência “Pórtico” contempla a monitorização das variáveis lógicas de entrada e saída da experiência, a partir de indicadores booleanos luminosos exibidos na janela de interface com o utilizador. A Tabela 4.2 contém a correspondência entre as etiquetas das variáveis partilhadas através do OPC Server (variáveis booleanas do PLC S7-1200) e o nome dos indicadores booleanos no painel frontal do sistema FDI, para a experiência “Pórtico”.

Tabela 4.2: Variáveis lógicas da experiência “Pórtico”.

Nome	Etiqueta da Variável Partilhada (S7-1200)
Electroíman	electroiman
Electroíman on	electroiman_on
Stop	stop
Avanço	av
Recuo	rec
Direita	dir
Esquerda	esq
Subir	sub
Descer	des
Reposicionamento	repo
Motor Avanço	MZav
Motor Recuo	MZrec
Motor Direita	MXdir
Motor Esquerda	MXesq
Motor Subir	MYsub
Motor Descer	MYdes
Sensor Avanço	SZav
Sensor Recuo	SZrec
Sensor Direita	SXdir
Sensor Esquerda	SXesq
Sensor Subir	SYsub
Sensor Descer	SYdes

O painel frontal e o diagrama de blocos do sistema FDI da experiência “Pórtico” encontram-se no Anexo B, página 125.

Electroíman Não Liga

Uma das possíveis falhas que podem ocorrer na experiência “Pórtico” é o facto da variável “electroiman_on” não ficar a 1 (não actuar) quando requisitada pela variável de entrada “electroiman”, isto é, do ponto de vista do utilizador o

electroíman não liga. A variável “electroíman” é a condição que estabelece se a saída “electroíman_on” fica a 1.

A fim de diagnosticar a falha detectada foram testadas as variáveis afectas ao processo ligar o electroíman: “electroíman_on”, “electroíman” e “SYdes”. Para chegar ao diagnóstico, a informação qualitativa foi expressa na forma de árvore de falhas com lógica booleana (ver Figura 4.10). Primeiramente, é realizado o teste booleano à variável “electroíman_on”, o valor booleano da variável (sintoma gerado) determina o ramo da árvore que deve ser seguido. Caso o resultado do teste seja 1 (Verdadeiro), o processo está a funcionar nas condições normais pois o electroíman efectivamente ligou. Pelo contrário, caso o sintoma gerado seja 0 (Falso), o ramo a seguir determina o teste a uma nova variável (“electroíman”).

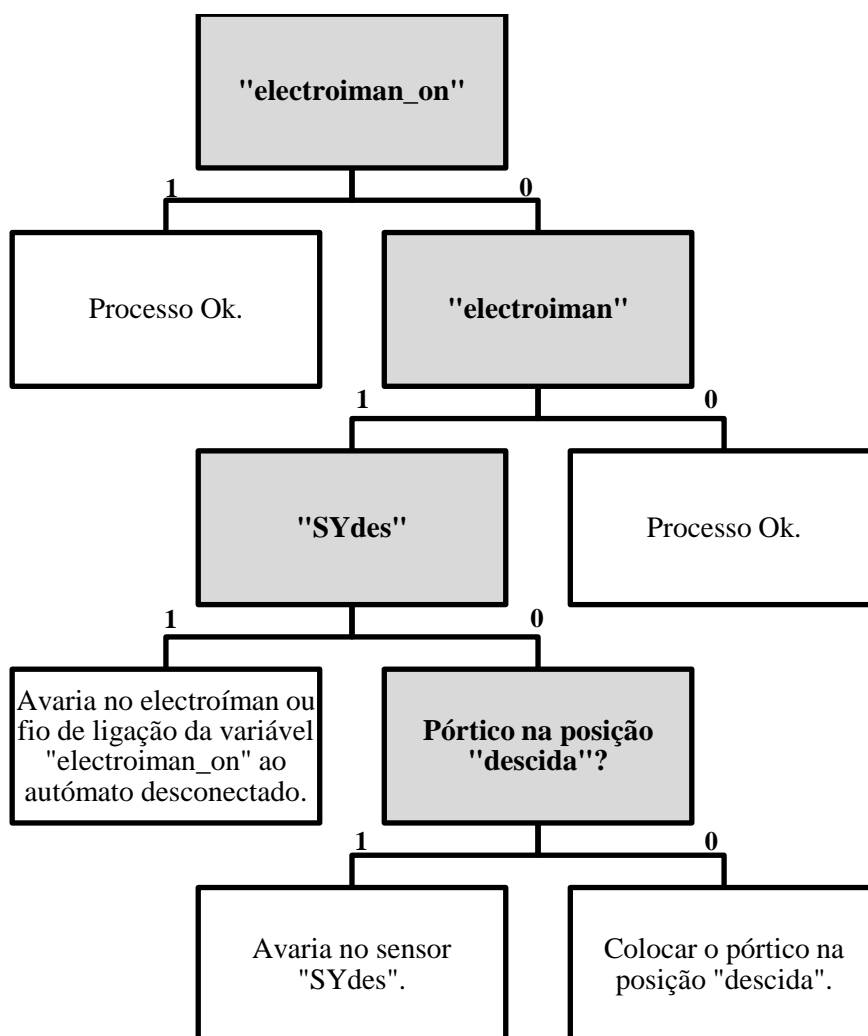


Figura 4.10: Árvore de falhas com lógica booleana para a detecção e diagnóstico de falhas do processo electroíman.

A variável “electroíman” é a condição que estabelece se a saída “electroíman_on” fica a 1, logo se o resultado do teste à variável “electroíman” for 0 significa que

não foi requisitado o accionamento da função electroíman, e o processo está novamente numa situação de funcionamento nominal. Por outro lado, se o valor booleano da variável “electroíman” for 1, o ramo da árvore de falhas determina o teste à variável “SYdes”.

A priori, sabe-se que o electroíman só liga se o pórtilco estiver na posição de “descida”, assim o sensor “descer” tem que estar a 1 para a variável “electroiman_on” ficar também a 1. Se o electroíman não liga mesmo com o “SYdes” a 1, ou seja, com o pórtilco na posição “descida”, a falha é diagnosticada como uma avaria no electroíman ou um fio de ligação da variável “electroiman_on” ao autómato desconectado. Caso o valor da variável “SYdes” seja 0, o utilizador é questionado acerca da posição do pórtilco (teste).

O sintoma gerado pelo teste “Pórtilco na posição “descida”?” determina se existe avaria no sensor “descer” (resposta afirmativa ao teste=1) ou indica ao utilizador que coloque o pórtilco na posição correcta (resposta negativa ao teste=0).

A aplicação da árvore de falhas, representada na Figura 4.10, através do LabVIEW foi feita com recurso a estruturas *Case* que como referido anteriormente, assemelham-se aos comandos “if, then, else, ...” de outras linguagens de programação porque permitem executar programaticamente secções de código gráfico mediante determinadas condições. De forma, a que fossem exibidos alarmes, a cada ocorrência de falha, na janela de interface com o utilizador, foi criado o sub-VI indicador de alarmes de cor e texto (“Alarme de cor e texto_electroiman”), ver Figura 4.12, que estabelece as cores do alarme exibido baseado no resultado dos testes às variáveis e devolve o texto a exibir no indicador de alarmes.

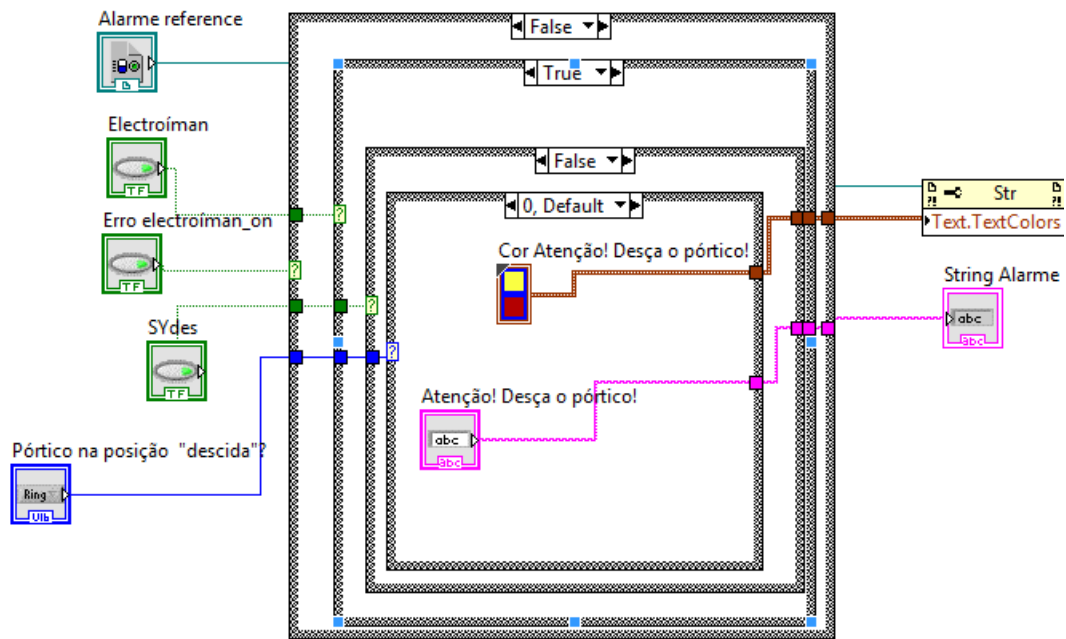


Figura 4.11: Diagrama de blocos do sub-VI para alarmes do processo electroímán.

A constituição do sub-VI é a seguinte:

1. Para cada teste da árvore de falhas é necessária uma *Case*, se existem quatro testes têm que existir quatro estruturas *Case*.
2. O valor de entrada ou terminal selector de cada *Case* é a variável a testar ou o teste à posição do pórtico. Cada uma das estruturas previstas tem dois subdiagramas ou casos. O terminal selector determina qual o caso a executar, mediante o sintoma gerado. Isto é, de forma análoga às regras IF-THEN se o valor de entrada for verdadeiro=1 (condição), então é executada a secção de código do subdiagrama da condição verdadeira (conclusão). Caso contrário se o valor de entrada for falso=0 (condição), então é executado o subdiagrama da condição falso (conclusão).
3. Os testes às variáveis “electroiman_on”, “electroiman” e “SYdes” são valores de entrada booleanos, representados por um controlo booleano. O teste “Pórtico na posição “descida”?” tem como terminal selector uma *Ring*. As *Rings* são objectos numéricos que associam valores numéricos a *strings* e têm a aparência de menu, no painel frontal, que o utilizador pode percorrer e fazer a sua selecção. A cada *string* corresponde um caso. Neste caso, a *Ring* tem associada as respostas à pergunta “Pórtico na posição “descida”?” em que “Sim” = 1 (Verdadeiro) e “Não” = 0 (Falso).

- Os subdiagramas contêm código gráfico que selecciona a cor e o texto adequados para o indicador do alarme.

O alarme de cor é conseguido através de um *Cluster*. Um *Cluster* é semelhante a um registo ou a uma estrutura, em linguagens de programação baseadas em texto, e consiste num grupo de elementos de dados que podem ter diferentes origens (numérica, booleana, texto, etc.), mas que não agrupa no mesmo *Cluster* controladores e indicadores. Neste caso, o *Cluster* agrupa dois indicadores numéricos de cor que permitem definir: a cor do texto do alarme e a cor de fundo do alarme. Caso ocorra uma falha o texto emitido é amarelo e o fundo do alarme vermelho (ver Figura 4.12), caso contrário o processo está a funcionar normalmente e o texto emitido apresenta cor preta e o fundo cor verde. O alarme de texto é realizado através de um controlo do tipo *String*, representado pela cor rosa no diagrama de blocos do sub-VI (ver Figura 4.11). Cada subdiagrama tem um controlador *String* diferente, consoante a falha ocorrida. Todos os controladores *String* têm um indicador único (comum), “String Alarme”, no exterior de todas as estruturas. O alarme de falha representado na Figura 4.12 é exibido na janela de interface com o utilizador, quando as variáveis “electroiman_on”, “electroiman” e “SYdes” estão a 1 e a resposta à pergunta “Pórtico está na posição “descida”?” é afirmativa (valor booleano=1).



Figura 4.12: Painel frontal do alarme de cor e texto do processo electroíman.

Por fim, os controlos booleanos, o menu *Ring* e o indicador “String Alarme” estão colocados no exterior de todas as estruturas *Case* para permitir o acesso aos mesmos aquando da chamada do sub-VI no VI principal “FDI”, conectando as variáveis aos terminais correspondentes.

No diagrama de blocos do VI principal “FDI” é invocado o sub-VI “Alarme de cor e texto_electroíman”, onde são ligados os terminais selectores de cada estrutura do sub-VI através das etiquetas das variáveis partilhadas (para as variáveis booleanas) e do menu *Ring* o teste à pergunta “Pórtico na posição “descida”?”. É também conectado o indicador *String* “Electroíman” que irá exibir os alarmes de falha no

painel frontal do VI principal. A Figura 4.13 exibe o diagrama de blocos da chamada do sub-VI para alarmes do processo electroíman no VI principal do sistema FDI.

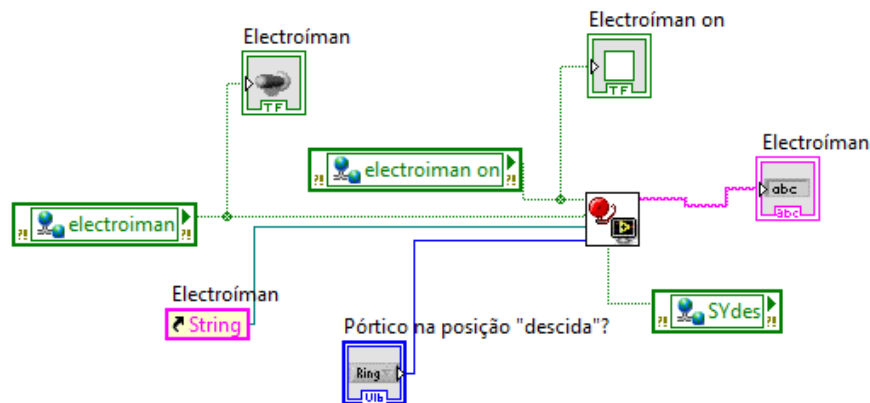


Figura 4.13: Chamada do sub-VI para alarmes do processo electroíman no VI principal do sistema FDI.

Pórtico Não Chega À Posição Pretendida

Outra das falhas possíveis de ocorrer na experiência “Pórtico” e prevista na secção 3.3.1. Arquitectura do Programa é o facto de o pórtico não chegar à posição requisitada pelo utilizador da experiência. Neste caso, definiu-se a monitorização da passagem do valor booleano do sensor da posição de partida do pórtico (sensor decisor) que irá despoletar o início de uma contagem de tempo e consequente verificação da ultrapassagem do limite fixo de tempo máximo que será o início da árvore de falhas (ver Figura 4.15) para a detecção e diagnóstico de falha no processo de movimento. Por exemplo, se o utilizador pretende que o pórtico execute o movimento de recuo, supõe-se que o sensor avanço esteja com valor 1. Assim, é testada a passagem de estado do sensor de 1 (verdadeiro) para 0 (falso), que irá iniciar uma contagem de tempo que apenas irá cessar quando o sensor de movimento contrário (recuo) passa de 0 para 1, ou caso o sensor de avanço passe de 0 para 1 (devido a uma requisição de avanço). Este processo é executado através de um sub-VI *Elapsed Time* construído para o efeito (ver Figura 4.14).

A verificação do limite fixo de tempo é realizada a partir do *express VI Elapsed Time*, este VI é usado para implementar um limite de tempo (*Time Out*). Quando o tempo que passou atinge a meta de tempo (*Time Target*) estabelecida, a variável de saída tempo passou (*Time Has Elapsed*) é verdadeira e é automaticamente feito o *reset* ao tempo, estando o *Elapsed Time* pronto para a nova contagem de tempo

quando a condição de entrada da *Case* seja novamente verdadeira. Para que o *express VI* só inicie mediante uma condição foi inserida uma *Case*, em que caso o valor de entrada da mesma seja verdadeiro (condição verdadeira) o tempo começa a correr. A condição para a execução do subdiagrama é então a passagem do estado booleano.

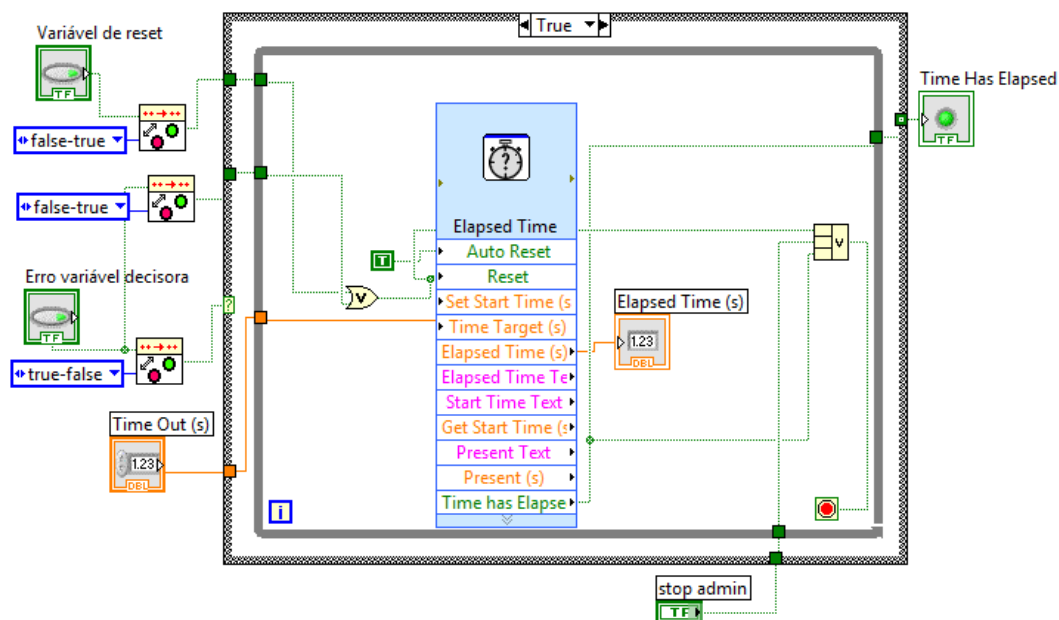


Figura 4.14: Diagrama de blocos do sub-VI *Elapsed Time*.

Voltando ao exemplo do processo de movimento de recuo do pórtilco, a verificação da passagem do estado booleano nos sensores referidos anteriormente, é feita com a função do LabVIEW *Boolean Crossing*. Em que, a entrada da função é o sensor de avanço, a direcção de detecção é do tipo verdadeiro-falso (de 1 para 0) e a passagem do estado booleano é o valor de entrada da estrutura *Case*. Caso o resultado da passagem de estado do sensor contrário (sensor recuo) de falso-verdadeiro (de 0 para 1) se verifique antes do tempo limite, é feito o *reset* ao *Elapsed Time* e o indicador *Times Has Elapsed* é falso (resultado do teste booleano). Aplicando a árvore de falhas da Figura 4.15, o sintoma gerado (*Times Has Elapsed=0*) traduz-se na informação de que o pórtilco chegou ao destino dentro do limite de tempo previsto e o processo encontra-se no seu funcionamento nominal. Se o resultado do teste ao indicador *Time Has Elapsed* for verdadeiro (1), o utilizador é questionado acerca da posição do pórtilco. Caso o sintoma gerado por este teste seja falso (0) a falha prevista é uma possível avaria no motor de recuo ou fio de ligação do motor de recuo ao autómato poderá estar desconectado, pois foi dada ordem de movimento de recuo e o motor tem saída 0. O ramo verdadeiro da

árvore determina que existe uma possível avaria no sensor de recuo ou que fio de ligação do sensor recuo ao autómato está desligado, devido ao pórtrico estar na posição pretendida (recuo) e o sensor apresentar saída 0.

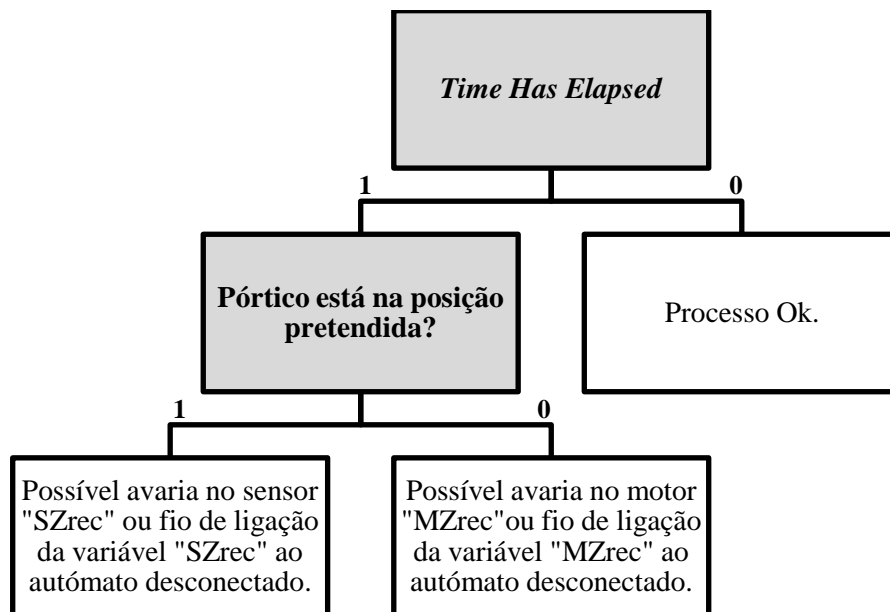


Figura 4.15: Árvore de falhas com lógica booleana para a detecção e diagnóstico de falhas do processo de movimento do pórtrico.

A árvore de falhas, representada na Figura 4.15, tipifica os alarmes de falha emitidos ao utilizador e que no LabVIEW é aplicada a partir do sub-VI indicador de alarmes de cor e texto (ver Figura 4.16), construído de forma análoga ao indicador de alarmes de falha no electroímã.

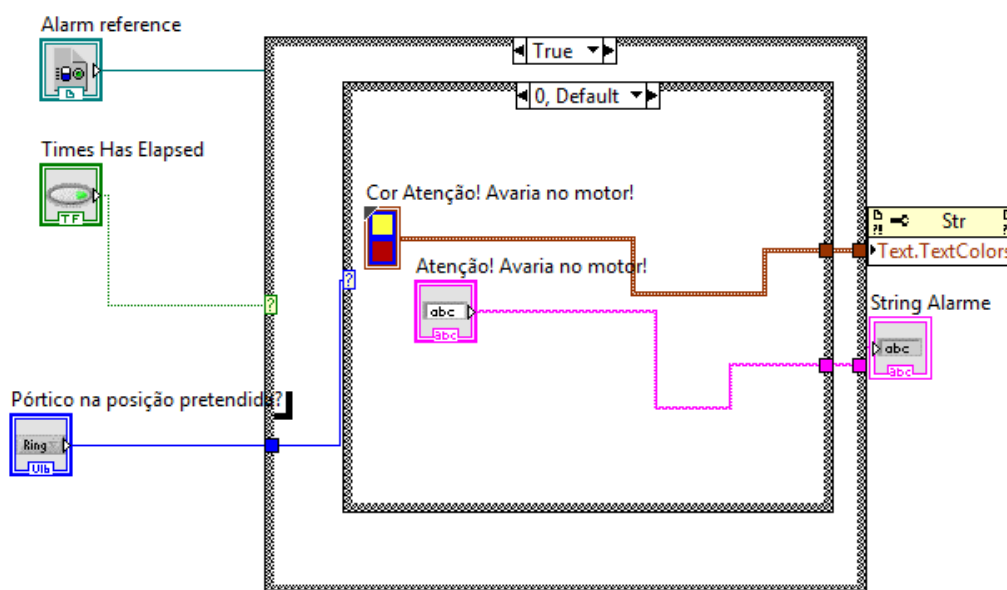


Figura 4.16: Diagrama de blocos do sub-VI para alarmes do processo de movimento do pórtrico.

Os sub-VIs *Elapsed Time* e alarmes de cor e texto são chamados ao VI principal onde são ligadas as variáveis lógicas da experiência necessárias para a detecção e o diagnóstico de falhas com consequente geração de alarmes (ver Figura 4.17). A saída do sub-VI *Elapsed Time* é o início da árvore de falhas aplicada ao sub-VI indicador de alarmes. As entradas situam-se do lado esquerdo e as saídas do lado direito dos sub-VIs.

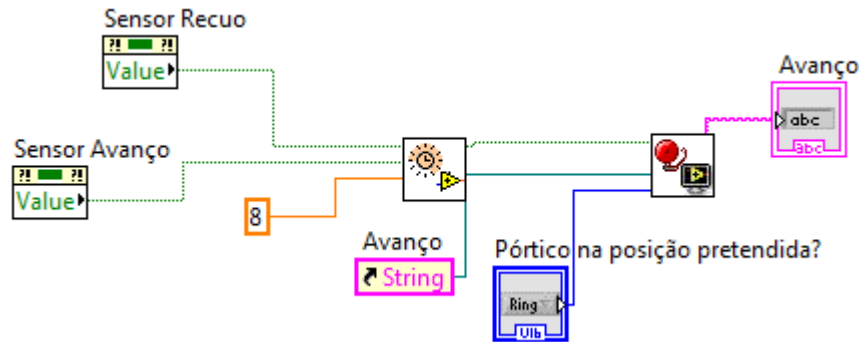


Figura 4.17: Chamada dos sub-VIs *Elapsed Time* e alarmes do processo avanço no VI principal do sistema FDI.

O raciocínio descrito anteriormente é aplicado para todos os movimentos do pórtico, podendo ser consultado o diagrama de blocos e o painel frontal com todas as variáveis monitorizadas e o *display* dos alarmes da experiência “Pórtico”, no Anexo B, página 125.

4.4. Experiência “Cilindros”

A experiência “Cilindros” consiste num painel de cilindros pneumáticos, representado na Figura 4.18, que é constituído por quatro cilindros pneumáticos e quatro válvulas solenoide que funciona em modo manual com o avanço e recuo de cada cilindro, e em modo automático, através de um ciclo (Ciclo 1) pré-definido.

O sistema FDI proposto aplicado à experiência “Cilindros” prevê a monitorização das variáveis lógicas de entrada e saída da experiência, a partir de indicadores booleanos luminosos exibidos na janela de interface com o utilizador, sendo também monitorizada a pressão da rede pneumática, com o auxílio de um transdutor de pressão.



Figura 4.18: Experiência “Cilindros”.

A Tabela 4.3 contém a correspondência entre as etiquetas das variáveis compartilhadas através do OPC Server (variáveis booleanas do PLC S7-1200) e o nome dos indicadores booleanos no painel frontal do sistema FDI, para a experiência “Cilindros”.

Tabela 4.3: Variáveis lógicas da experiência “Cilindros”.

Nome	Etiqueta da Variável Lógica do PLC
I0.0	input_0
I0.1	input_1
I0.2	input_2
I0.3	input_3
I0.4	input_4
I0.5	input_5
I0.6	input_6
I0.7	input_7
Q0.0	output_0
Q0.1	output_1
Q0.2	output_2
Q0.3	output_3
Q0.4	output_4
Q0.5	output_5
Q0.6	output_6
Q0.7	output_7
Ciclo 1	ciclo_1
Cilindro A	cilindro A
Cilindro B	cilindro B
Cilindro C	cilindro C
Cilindro D	cilindro D
Modo Manual	modo_manual
Modo Automático	modo_auto
Emergência	stop

O painel frontal e o diagrama de blocos do sistema FDI da experiência “Cilindros” encontram-se no Anexo B, página 125.

4.4.1. Pressão da Rede Pneumática

A rede pneumática alimenta o painel de cilindros da experiência “Cilindros” do LabAPI – ISEL. No sistema proposto, a pressão desta rede é monitorizada ao longo

do tempo utilizando um transdutor de pressão com uma gama de medida de 0 a 10 bar e uma saída em corrente de 4 a 20 mA. O sinal analógico resultante da medição do sensor é digitalizado através da placa de aquisição de dados NI USB-6008 (ver Figura 3.3, página 51) e da *driver* de comunicação entre a placa DAQ e o programa do sistema proposto, NI-DAQmx. A *driver* NI-DAQmx previne que se tenha que recorrer a programação de baixo nível e a programação de comandos complicados de forma a ter acesso às funções do *hardware*. A conexão entre a NI USB-6008 e o computador onde o LabVIEW corre é feita a partir de uma porta USB. Para a medição simples é usado o sinal AI 0 da placa DAQ (ver Figura 3.4, página 52).

O sinal digitalizado pela DAQ é obtido no diagrama de blocos pelo *express VI DAQ Assistant*, em que a saída de dados é visualizada no painel de interface com o utilizador a partir de indicadores do tipo numérico, gráfico (que exhibe a visualização da pressão ao longo tempo) e analógico.

A Figura 4.19 representa o diagrama de blocos do processo de monitorização e detecção de anomalias na pressão da rede pneumática.

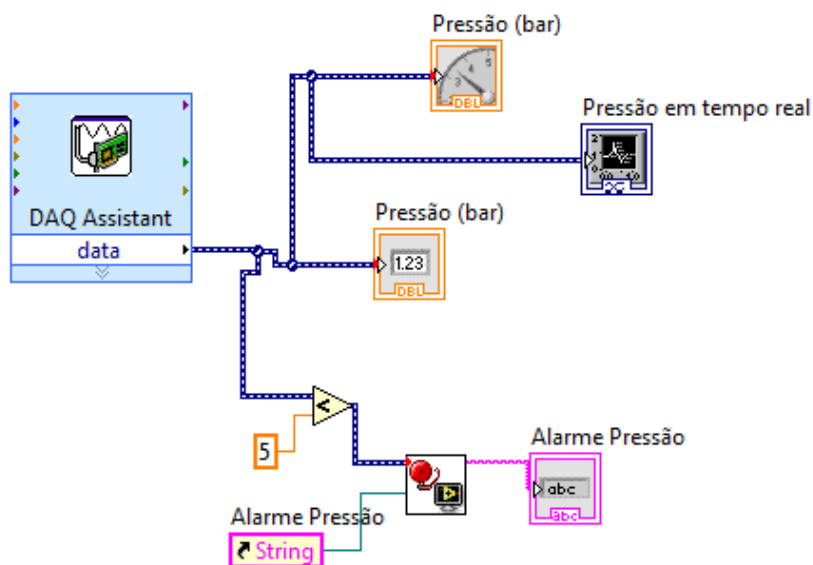


Figura 4.19: Diagrama de blocos do processo de detecção e diagnóstico de falhas da pressão da rede pneumática.

O teste à variável analógica é realizado com recurso à verificação do limite mínimo de pressão requerida para as cilindros actuarem de forma a cumprirem os movimentos pedidos pelo utilizador sem interrupções ou atrasos, tendo sido estabelecido o limite mínimo de 5 bar. No diagrama de blocos este teste é executado através de funções de comparação que, caso a pressão medida naquele

instante seja inferior ao limite mínimo aceite, o sistema emite um alarme visual de indicação de pressão na rede ineficaz e alertando para a necessidade de contactar o administrador: “ATENÇÃO: Fuga na rede pneumática ou compressor desligado. Por favor, encerre a sessão e contacte o administrador.”. Pelo contrário, caso a pressão esteja acima do valor mínimo permitido, o sistema dará a indicação ao utilizador de “Pressão da rede ok”. O sub-VI alarme de cor e texto descrito em pormenor na secção 4.3. Experiência “Pórtico” possibilita a emissão dos alarmes supra mencionados.

4.4.2. Variáveis Lógicas

Cilindro Não Avança/Não Recua

Na experiência “Cilindros” previu-se a ocorrência de falha no movimento de avanço/recuo dos cilindros pneumáticos. Partindo do exemplo da não execução do movimento de avanço do cilindro A, definiu-se tal como na detecção de falha de movimento do pórtico requerido pelo utilizador, na secção 4.3. Experiência “Pórtico”, a monitorização da passagem do valor booleano que irá iniciar o sub-VI *Elapsed Time_Cilindros* (ver Figura 4.20). Contudo, neste caso a variável a monitorizar será a “cilindro A” que quando a 1, origina a saída da variável “output_0” (variável de sinal de avanço da válvula solenoide). O tempo limite verificado é então o tempo que o cilindro pneumático demora a chegar ao fim de curso. O sub-VI *Elapsed Time_Cilindros* é cessado quando a variável “cilindro A” para de 1 para 0, isto é, quando o cilindro pneumático chega ao fim de curso, ou quando o limite de tempo implementado atinge a meta de tempo, o que resulta na condição verdadeira do indicador *Time Has Elapsed*.

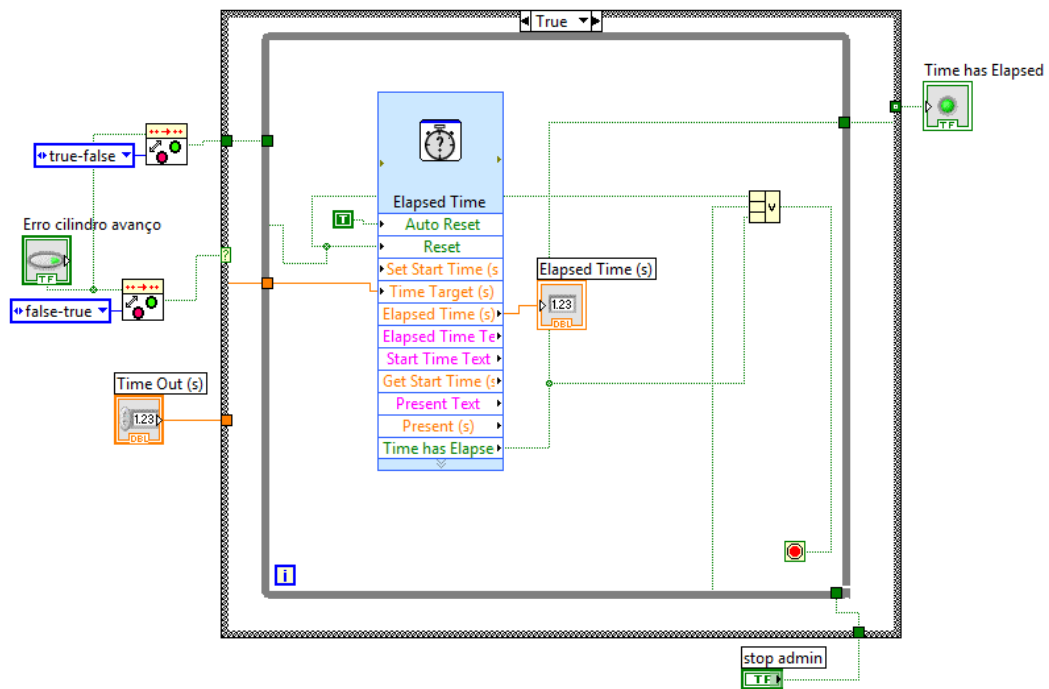


Figura 4.20: Diagrama de blocos do sub-VI *Elapsed Time_cilindros*.

Aplicando a árvore de falhas da Figura 4.21, o sintoma gerado (*Times Has Elapsed=0*) traduz-se na informação de que o cilindro chegou à posição de avanço dentro do limite de tempo previsto e o processo encontra-se no seu funcionamento nominal. Se o resultado do teste ao indicador *Time Has Elapsed* for verdadeiro (1), o ramo de decisão segue o teste à variável “modo_manual”. O cilindro só avança se esta condição for verdadeira, se for falsa o sistema alerta o utilizador para que ligue o modo manual da experiência “Cilindros”.

Todavia, se esta condição já se verificar o ramo da árvore determina o teste à variável de fim de curso de avanço “input_1” e o sintoma gerado notifica o utilizador caso a variável já esteja a 1, isto é, caso o cilindro já esteja no avanço prevenindo lapsos por parte do utilizador. O resultado falso do teste à variável “input_1” origina o teste à variável “output_0”. Que estando actuada deveria significar o avanço do cilindro pneumático, num entanto se ocorrer uma avaria no cilindro ou na própria válvula o cilindro não avança. Por outro lado, se o sintoma gerado pelo teste à variável de saída da válvula for falso (0) o utilizador é questionado de modo a aferir o estado da pressão da rede pneumática que pode ser monitorizada no painel frontal. Esta pergunta é respondida através de um botão do tipo controlador booleano que caso seja accionado traduz-se numa resposta negativa (falsa) originando o resultado de uma possível fuga na rede pneumática ou que o compressor encontra-se desligado. O teste “Pressão ok?” podia ser

realizado automaticamente através da ligação sub-VI de geração de alarmes da pressão da rede pneumática. Porém, este método permite uma maior interacção com o utilizador remoto.

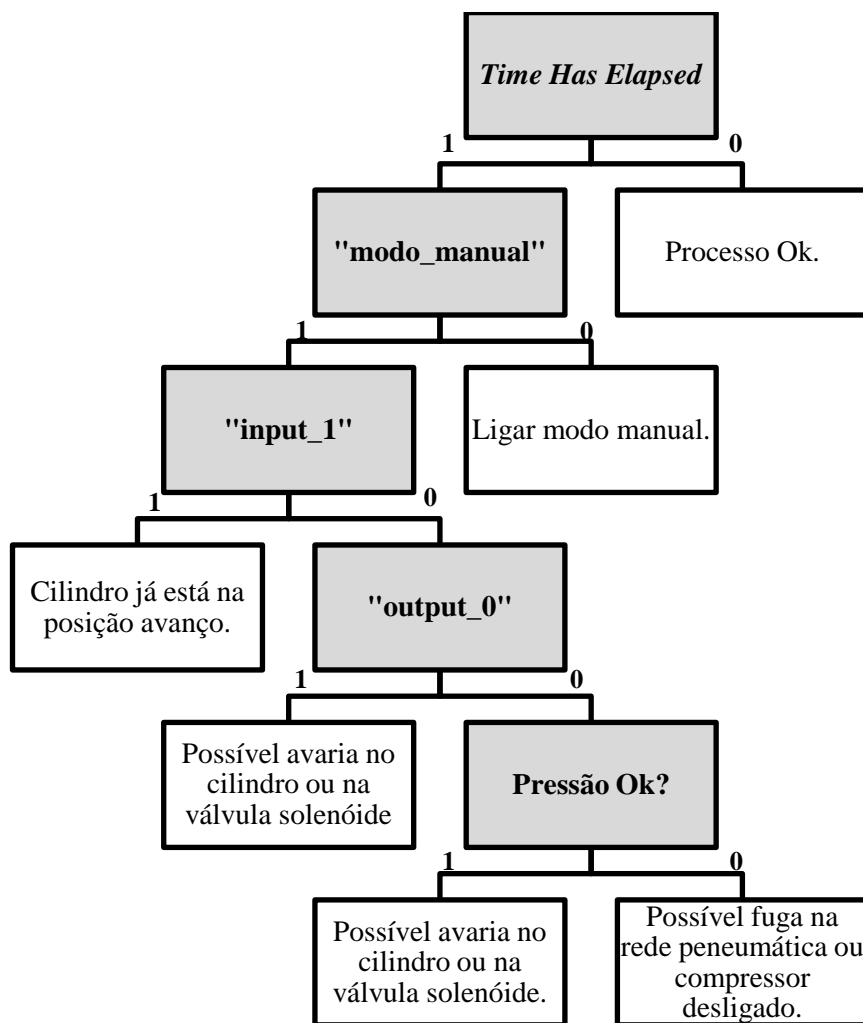


Figura 4.21: Árvore de falhas com lógica booleana para a detecção e diagnóstico de falhas do processo cilindro não avança/não recua.

A árvore de falhas representada na Figura 4.21/ Figura 4.15 tipifica os alarmes de falha emitidos ao utilizador sendo aplicada a partir do sub-VI indicador de alarmes de cor e texto de estrutura semelhante à descrita na secção 4.3. Experiência “Pórtico” na falha “Pórtico Não Chega Posição Pretendida” .

Os sub-VIs *Elapsed Time_cilindros* e alarmes de cor e texto são chamados ao VI principal onde são ligadas as variáveis lógicas da experiência necessárias para a detecção e o diagnóstico de falhas com consequente geração de alarmes (ver Figura 4.22). A saída do sub-VI *Elapsed Time_cilindros* é o início da árvore de falhas aplicada ao sub-VI indicador de alarmes. As entradas situam-se do lado esquerdo e as saídas do lado direito dos sub-VIs.

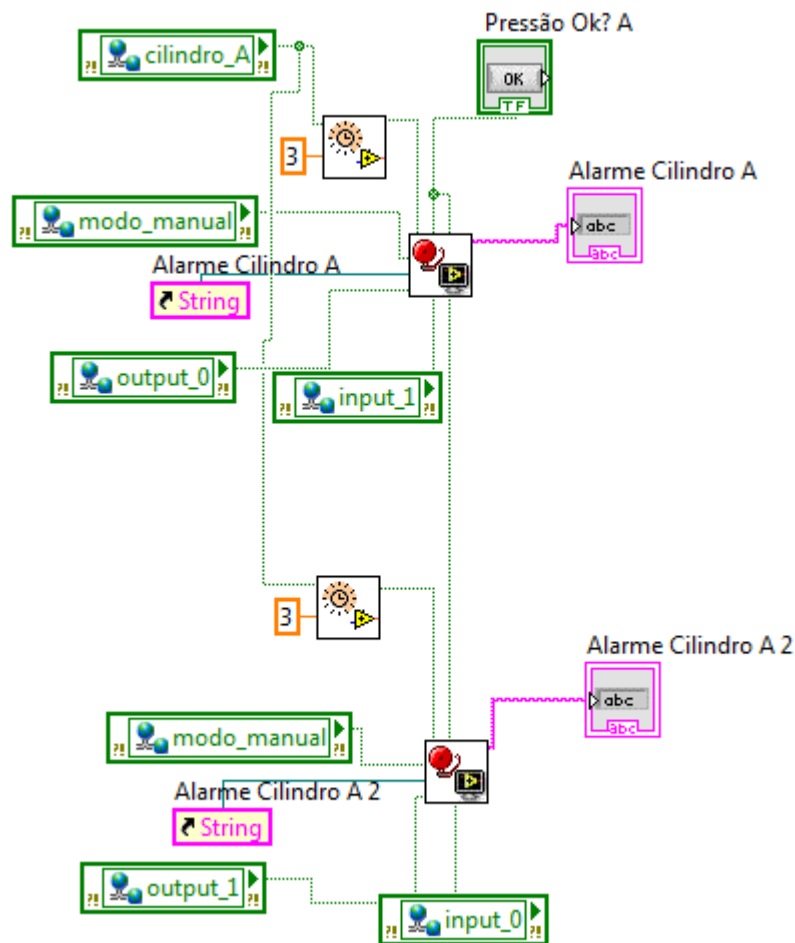


Figura 4.22: Chamada dos sub-VIs *Elapsed Time_Cilindros* e alarmes do processo avanço e recuo no VI principal do sistema FDI.

O raciocínio descrito anteriormente é aplicado para todos os cilindros, podendo ser consultado o diagrama de blocos e o painel frontal com todas as variáveis monitorizadas e o display dos alarmes da experiência “Cilindros”, no Anexo B, página 125.

Ciclo 1 Não Inicia

A monitorização e consequente detecção que o Ciclo 1 não inicia a sequência de movimento de cilindros segue a mesma lógica de aplicação de uma árvore de falhas como elemento decisor do tipo de falha ocorrida (ver Figura 4.23) e utiliza o sub-VI *Elapsed Time_cilindros* sendo que o valor de entrada da estrutura *Case* que determina o início da contagem do tempo limite é a passagem do estado booleano da variável “ciclo 1”. Caso o indicador booleano *Time Has Elapsed* seja verdadeiro (1), é realizado o teste à variável “modo_auto” que é uma das condições

para que o ciclo 1 tenha início. O sintoma gerado por este teste leva a novos ramos de decisão onde na situação em que o modo automático esteja efectivamente ligado e o ciclo mesmo assim não inicie, o utilizador é inquerido sobre a pressão da rede pneumática visível através dos indicadores de pressão da rede no painel frontal. A resposta do utilizador traduz-se num resultado do teste com dois tipos diferentes de falhas.

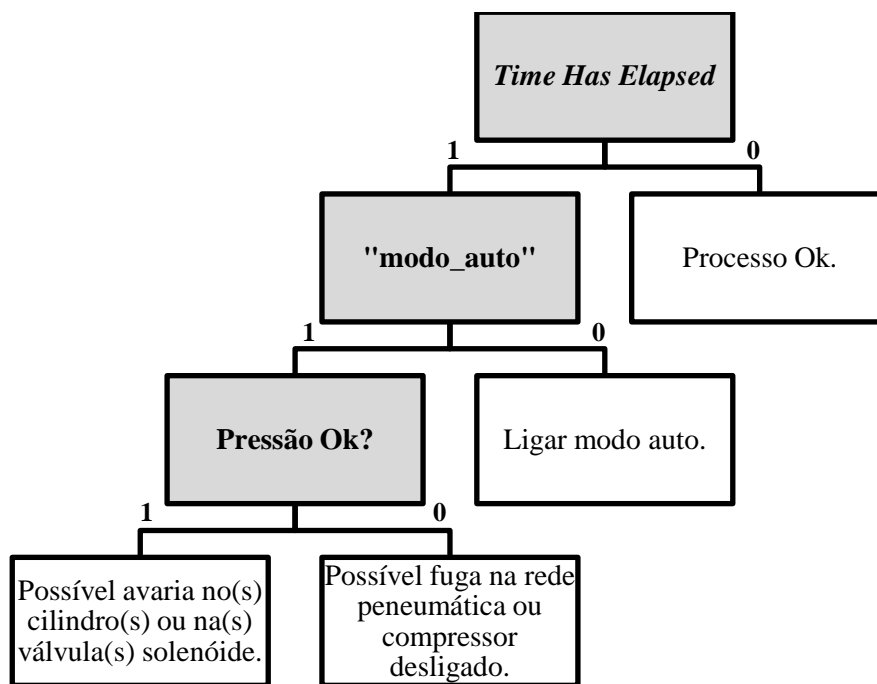


Figura 4.23: Árvore de falhas com lógica booleana para a detecção e diagnóstico de falhas do processo ciclo 1.

A inserção dos sub-VIs Elapsed Time_cilindros e alarmes do processo ciclo 1 no VI principal é realizada da mesma forma diferenciando apenas no valor da constante de limite de tempo (11 segundos).

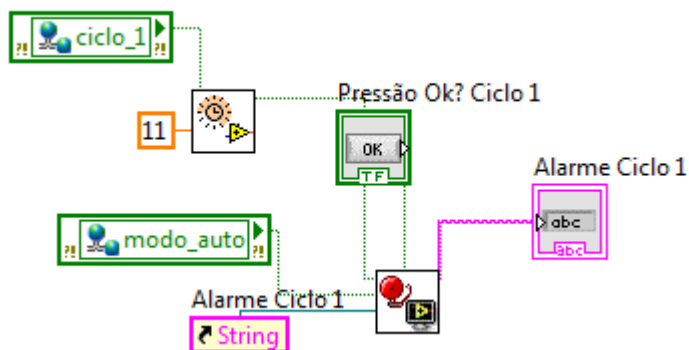


Figura 4.24: Chamada dos sub-VIs *Elapsed Time_Cilindros* e alarmes do processo ciclo 1 no VI principal do sistema FDI.

4.5. Análise de Vibrações

A configuração do sistema de análise de vibrações proposto está exemplificado na Figura 4.25. Genericamente, este é dividido em vários módulos e contém: aquisição e simulação de sinal por parte do utilizador, exibição de ondas, análise de sinal no domínio do tempo, e no domínio da frequência, conjuntamente com outras particularidades. Os parâmetros sensíveis ao dano utilizados para a análise de vibrações são o sinal no tempo, espectro FFT, o nível global e o factor de crista. O utilizador pode escolher o tipo de sensor que pretende utilizar para a aquisição de sinal, bem como a unidade de entrada e saída, e este pode ainda ser calibrado. O sistema prevê o alerta para a condição do equipamento através de alarmes.

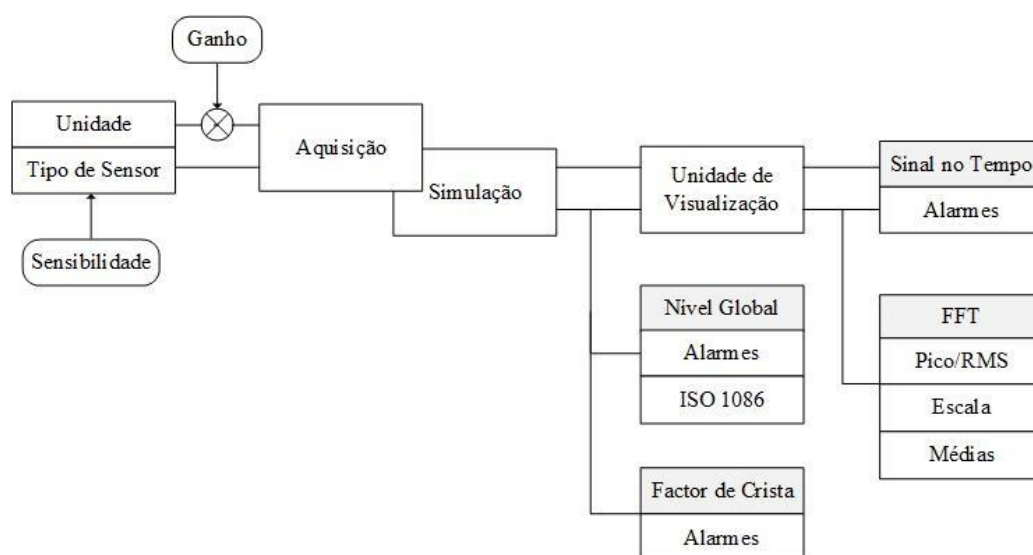


Figura 4.25: Esquema da configuração do sistema de análise de vibrações.

Tendo em consideração que o sistema de monitorização e detecção de vibrações é testado num laboratório remoto com o objectivo educacional, definiu-se que seria vantajoso para o utilizador não só poder analisar os parâmetros do sinal adquirido, como também poder simular um sinal através da escolha de alguns elementos que caracterizam os sinais harmónicos.

4.5.1. Configuração dos Dados de Aquisição

No módulo de configuração dos dados de aquisição definem-se as características da aquisição de dados. A Figura 4.26 exhibe a estrutura do painel de aquisição de dados e opções possíveis.

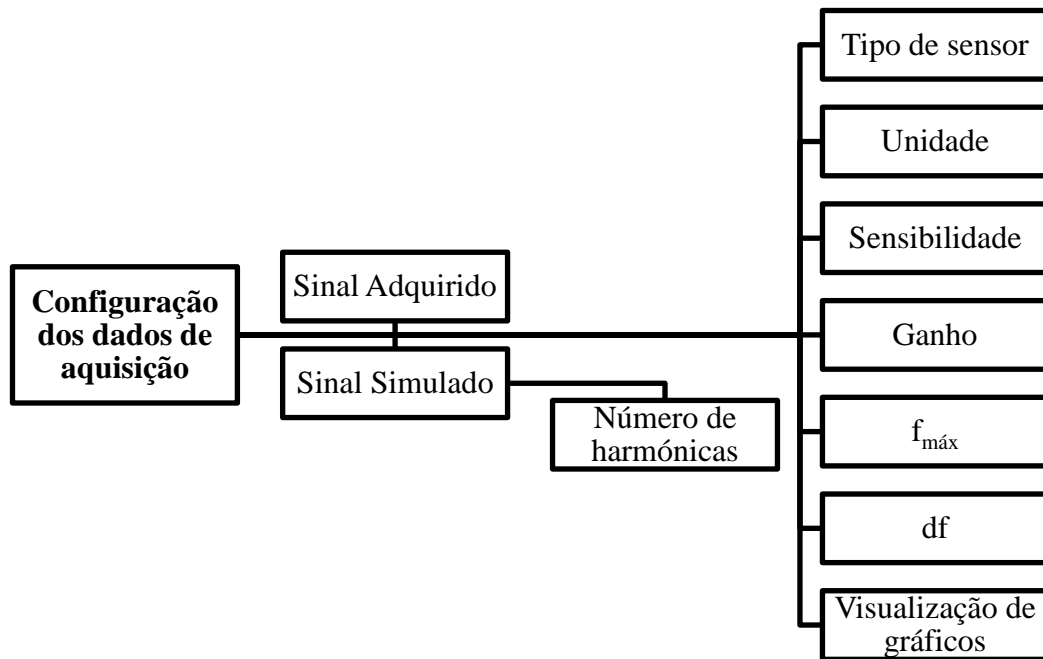


Figura 4.26: Fluxograma da estrutura do módulo de configuração dos dados de aquisição.

A configuração dos dados de aquisição é feita primeiramente pela escolha do tipo de sinal: sinal adquirido ou sinal simulado. Esta escolha é possível através de um botão booleano “Sinal Simulado/Sinal Adquirido”, disponível na janela de interface com o utilizador, que controla a estrutura *Case* que executa o código correspondente a cada subdiagrama.

Para o utilizador poder simular um sinal é necessário que este escolha alguns elementos como a selecção de uma, duas ou três harmónicas consoante o pretendido pelo utilizador. O sinal é simulado através do *express VI Simulate Signal* (ver Figura 4.27), que simula ondas sinusoidais, ondas quadradas, ondas triangulares ou sinais de ruído através da introdução das propriedades amplitude, frequência e fase, que no caso do sistema proposto são inseridas pelo utilizador em qualquer uma das situações (uma, duas ou três harmónicas), através de controladores numéricos. Para cada harmónica é necessário um *Simulate Signal*. A selecção do número de harmónicas é também providenciada através de uma estrutura *Case* controlada por uma *Ring* “Nº de Harmónicas”. Além destas propriedades são introduzidas outras imprescindíveis para uma correcta aquisição de sinal: o tipo de sensor, a respectiva unidade de medição e a sensibilidade do sensor, sendo também possível a introdução da propriedade ganho para efeitos de calibração do sensor.

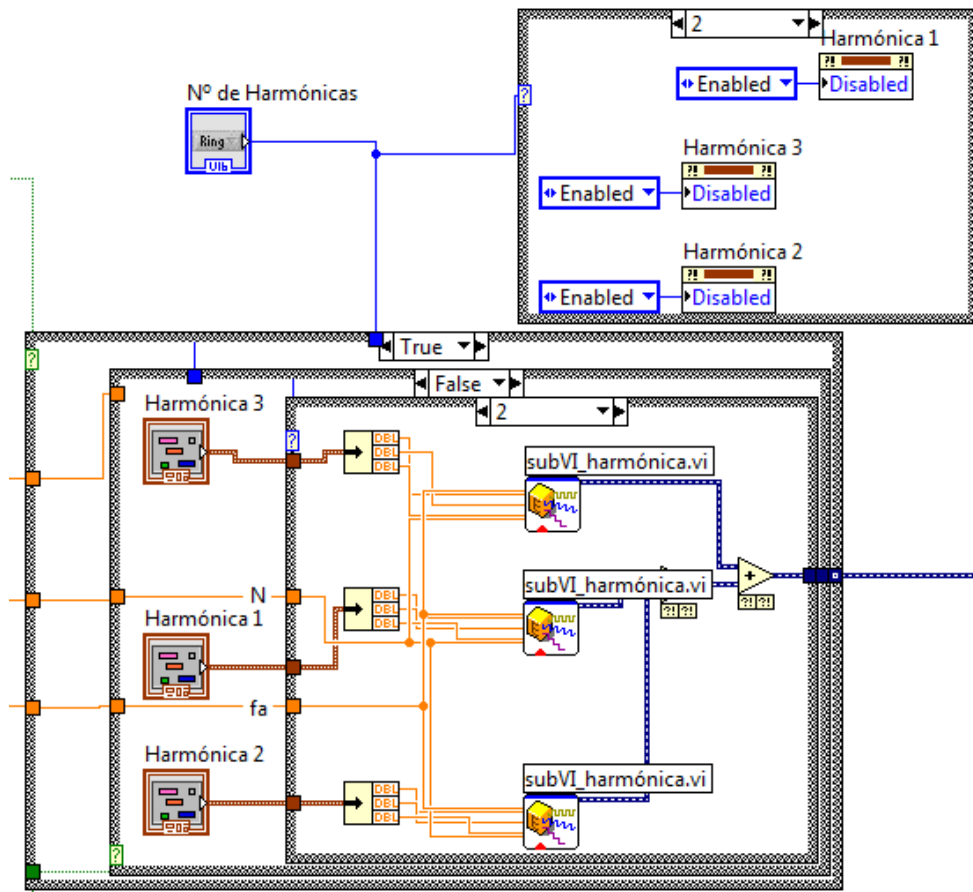


Figura 4.27: Simulação de sinal através de um *express VI* no diagrama de blocos.

Por outro lado, o modo sinal adquirido mede o sinal através do *express VI Acquire Sound* que utiliza a placa de som do computador como placa de aquisição (ver Figura 4.27). Para o *Acquire Sound* adquirir os dados necessários precisa da indicação da resolução em frequência (df) e da frequência de amostragem (f_a) do sinal. É no módulo de configuração dos dados de aquisição que o utilizador insere os valores da frequência mais alta que será capturada e exibida nos vários módulos do analisador de vibrações ($f_{máx}$) e da resolução em frequência (df) através de controladores numéricos. Em simultâneo é indicada a frequência de amostragem (f_a), o número de pontos a medir (N) e o período de amostragem (T), que são calculados através dos elementos $f_{máx}$ e df . Estes elementos podem ser visualizados na Figura 4.28, que exhibe a configuração da estrutura *Case* na situação sinal adquirido, os controladores e indicadores numéricos, e as operações matemáticas necessárias para a obtenção dos parâmetros df e f_a .

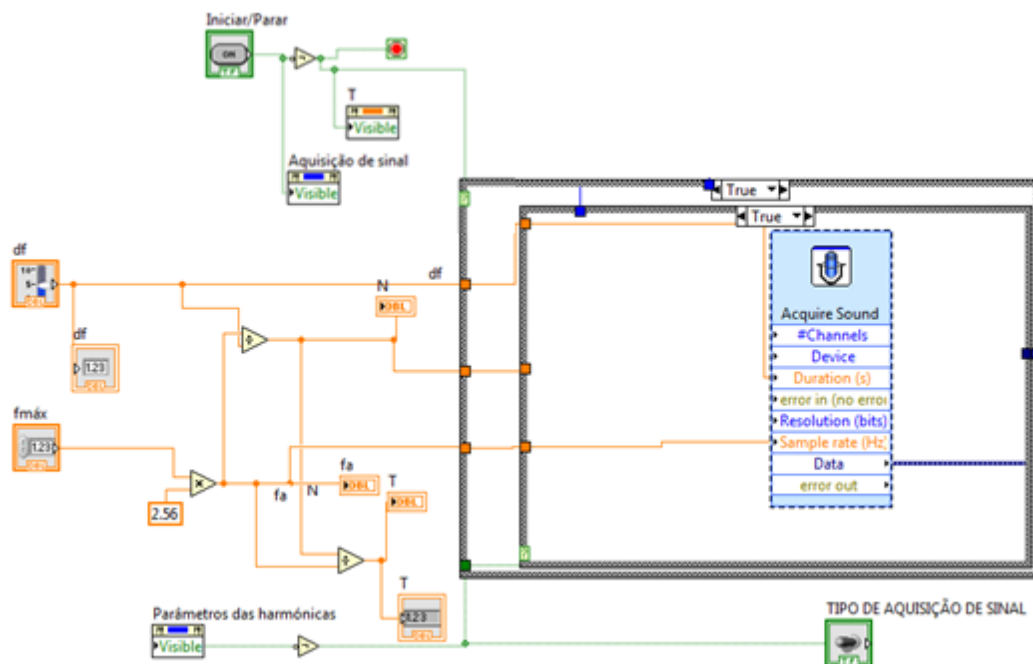


Figura 4.28: Aquisição de sinal através do express VI *Acquire Sound* no diagrama de blocos.

O programa adequa-se à utilização de sensores de deslocamento, velocidade e aceleração, e para cada tipo de sensor disponível, é possível medir vibrações em mais do que uma unidade de medida. A unidade de medida depende do tipo de entrada de sinal escolhido, deste modo no sistema proposto contemplam-se as seguintes:

- Deslocamento – mv/ μm e mv/mm;
- Velocidade – mv/($\mu\text{m/s}$) e mv/(mm/s);
- Aceleração – mv/(mm/s²) e mv/g.

Escolhido o sensor, e para utilizá-lo é imprescindível calibrá-lo. A calibração é feita medindo um sinal conhecido, por exemplo 10 m/s² a 150 Hz. Se for medido 5 m/s² nos 150 Hz então é necessário calibrar multiplicando o sinal por 2 (ganho). Assim, o sistema desenvolvido prevê esta propriedade para que o utilizador possa calibrar o seu sensor.

O módulo de configuração dos dados de aquisição pode ser visualizado na sua totalidade, da perspectiva do utilizador e em modo sinal simulado, na Figura 4.29.

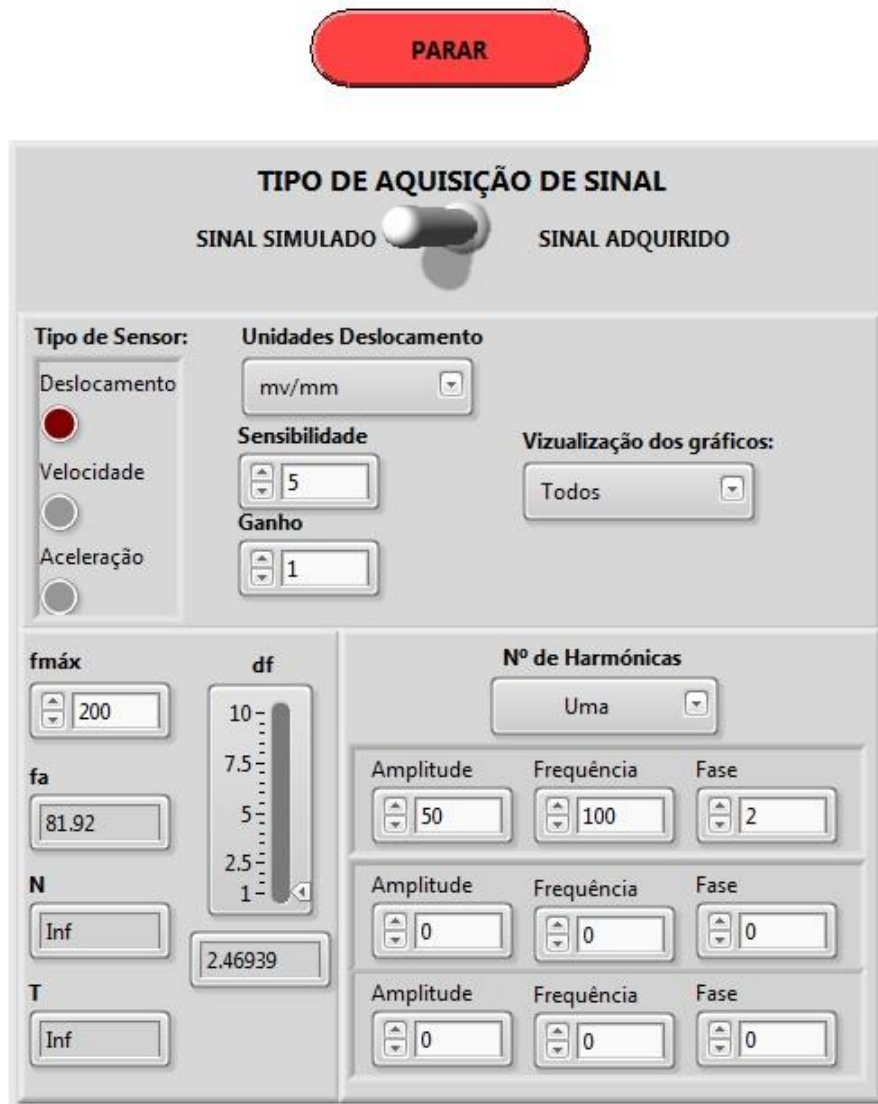


Figura 4.29: Painel frontal do módulo de configuração dos dados de aquisição do sistema de análise de vibrações em modo simulado

A partir dos dados de aquisição definidos anteriormente no módulo de configuração dos dados de aquisição obtêm-se os diferentes parâmetros sensíveis ao dano: sinal no tempo, espectro FFT, factor de crista e nível global. No painel de aquisição de sinal o utilizador tem a opção de seleccionar quais os parâmetros que se querem visualizar através de uma *Ring* do tipo menu, podendo escolher-se entre: todos, sinal no tempo e espectro FFT, sinal no tempo e nível global, ou sinal no tempo e factor de crista. A invisibilidade dos módulos correspondentes aos parâmetros de análise das vibrações medidas no painel frontal, é exequível com recurso a *Property Nodes* e a uma *Case*, no diagrama de blocos (ver Figura 4.30). O *Property Node* obtém e define propriedades de uma referência em aplicações locais ou remotas, em VIs, e em objectos. Em relação à selecção e correspondente invisibilidade dos parâmetros sensíveis ao dano é utilizado um *Property Node* do

tipo *Visible* gerido a partir de um controlo booleano *T/F – True/False* (V/F – Verdadeiro/Falso).

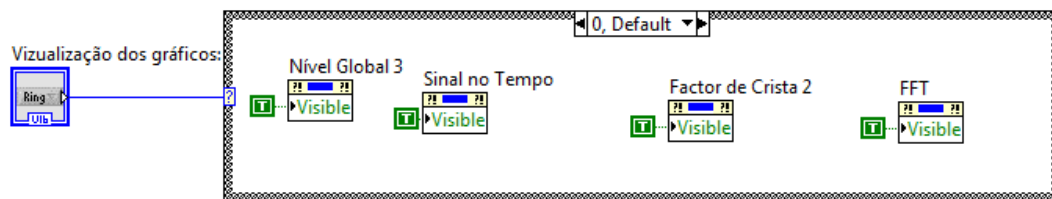


Figura 4.30: Property Nodes, no diagrama de blocos, para a visibilidade/invisibilidade dos diversos gráficos.

Por último, referir que o módulo de configuração dos dados de aquisição apenas pode ser acedido quando não ocorre medição de sinal porque ao alterar o tempo de aquisição ou o número de leituras durante a aquisição de sinal, não se iriam obter valores de histórico comparáveis. De forma a impossibilitar o acesso aos dados de aquisição quando o analisador de vibrações está a medir sinal, é contemplada a existência de uma estrutura do tipo *Case* no diagrama de blocos, como é visível na Figura 4.28, sendo controlada com recurso a um botão do tipo booleano “Iniciar/Parar”, em ambos os ambientes LabVIEW.

4.5.2. Sinal no Tempo

O sinal adquirido pelos diferentes tipos de aquisição de dados (sinal simulado ou sinal adquirido) é analisado no domínio do tempo no módulo “Sinal no tempo” (ver Figura 4.31).

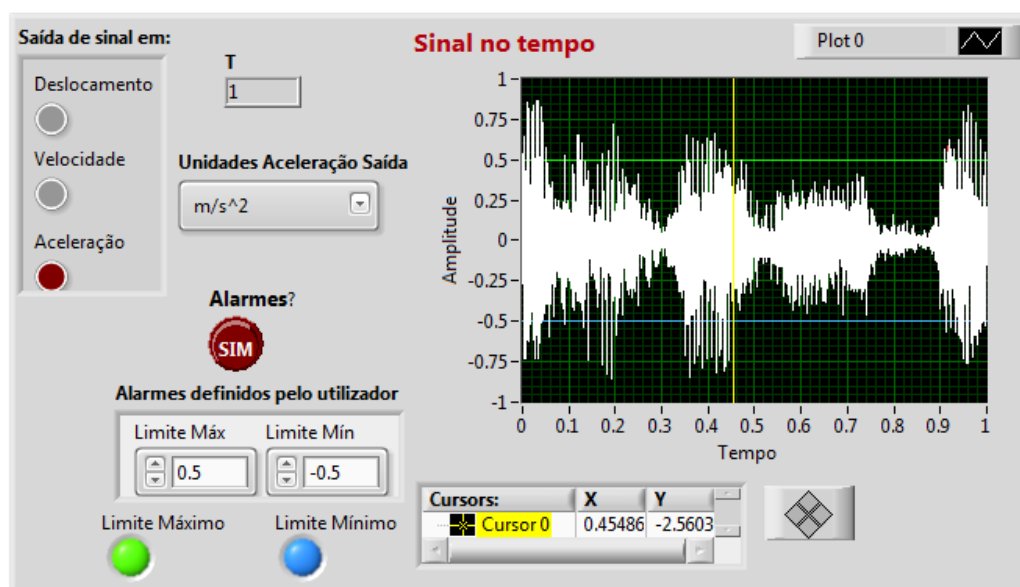


Figura 4.31: Painel frontal do módulo sinal no tempo.

O utilizador pode visualizar o sinal no tempo em deslocamento, velocidade ou aceleração. A unidade de medida depende do tipo de saída de sinal escolhido, deste modo no sistema proposto contemplam-se as seguintes:

- Deslocamento - μm , mm e m;
- Velocidade - $\mu\text{m/s}$, mm/s e m/s;
- Aceleração - $\mu\text{m/s}^2$, mm/s², m/s² e g.

Como referido em 2.4.1. Conceitos de Vibração, o deslocamento, a velocidade e aceleração estão relacionadas através de funções simples de derivação ou integração. Assim, independentemente do sensor utilizado (deslocamento, velocidade ou aceleração) para a aquisição, o sinal pode ser processado directamente ou passar por processos matemáticos de conversão para outras unidades de medida de vibrações. Deste modo, o sinal gerado a partir das configurações de aquisição de dados, no módulo correspondente, é conectado a várias estruturas *Case* responsáveis pelas integrações e derivações do tipo de entrada e de saída dos sensores, como se verifica na Figura 4.32.

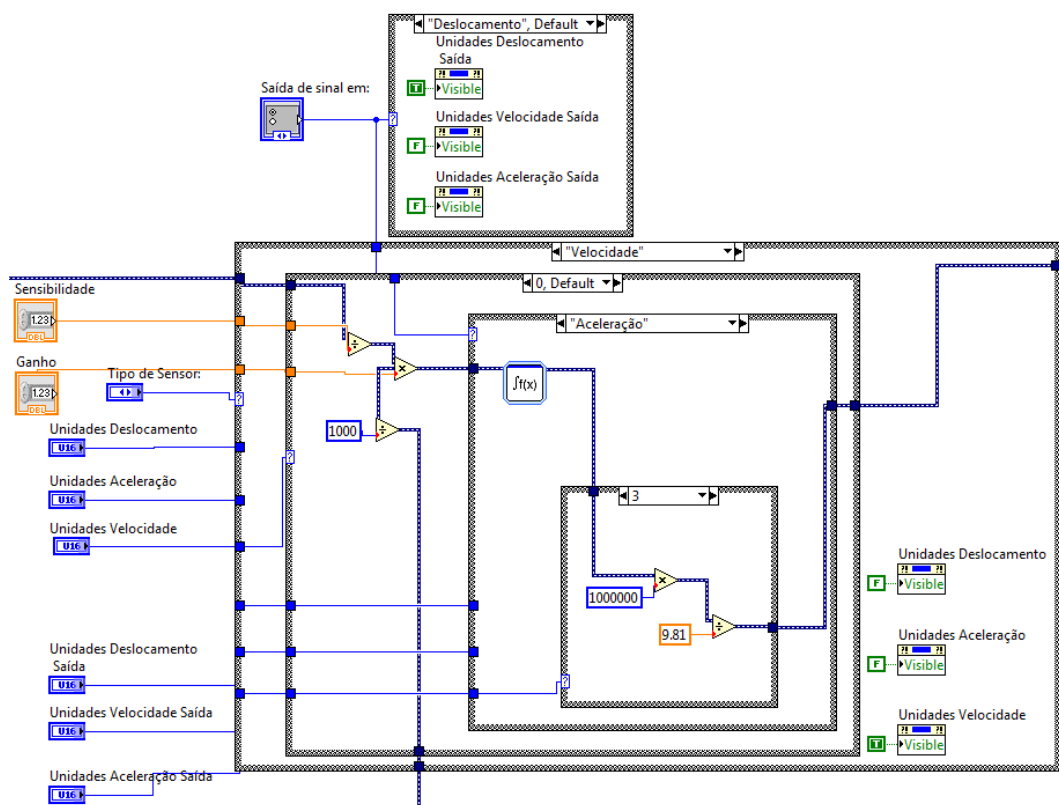


Figura 4.32: Estruturas Case das conversões de sinal no diagrama de blocos.

Quando se recorre a uma função de integração, por exemplo quando a entrada de sinal é em aceleração e se pretende visualizar o sinal de saída em velocidade, esta é imediatamente precedida de um filtro do tipo passa-alto, caso contrário o sinal será corrompido pela constante de integração C_1 , (ver (4.1)), originando ruído e sinais vibratórios que se encontram fora da banda de frequência de interesse:

$$a = \int \omega(z)dz + C_1 \quad (4.1)$$

Se existirem duas integrações sucessivas, por exemplo quando a unidade de entrada é de aceleração e a unidade de saída é de deslocamento, é necessário condicionar o sinal com dois filtros do tipo passa-alto, um após cada integração. Os filtros passa-alto são filtros de ondas que têm uma banda de transmissão única que se estende de alguma frequência crítica ou frequência de corte, diferente de zero, até à frequência infinita e têm como objectivo eliminar o ruído das baixas frequências [45, 48]. Neste caso, é determinada uma frequência de corte padrão de 100 Hz (ver Figura 4.33), através do *express VI Filter* (filtro) do tipo *Highpass*.

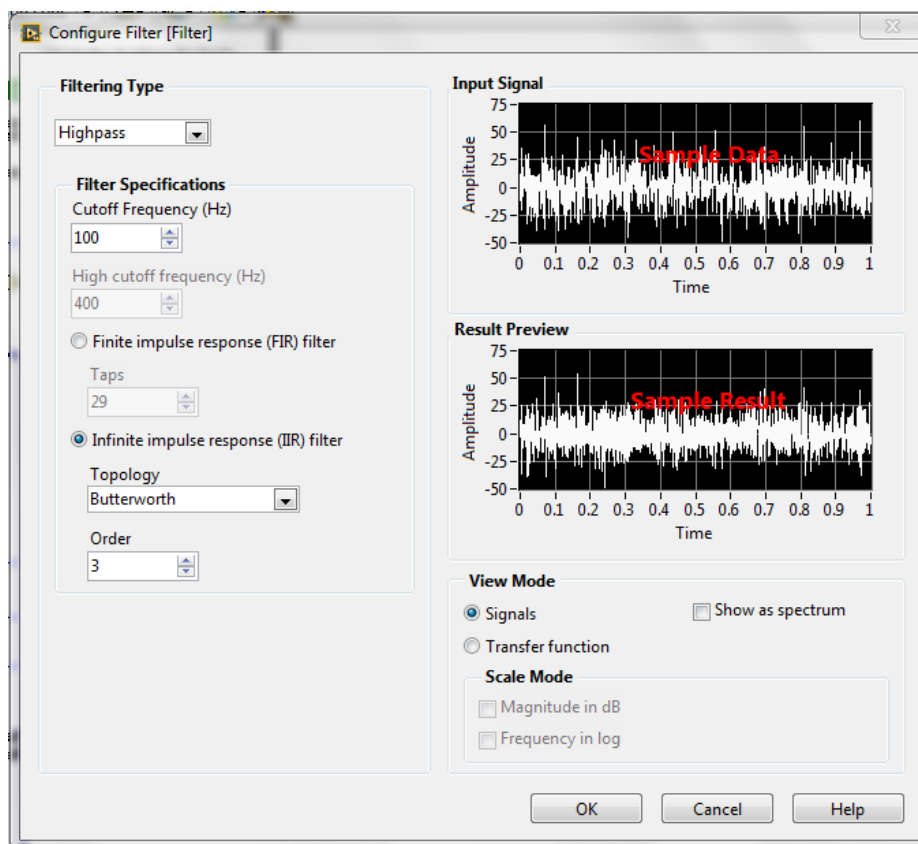


Figura 4.33: Configuração do filtro *Highpass* (passa-alto) no diagrama de blocos.

A partir desta estrutura Case “Deslocamento/Velocidade/Aceleração”, o sinal medido sai para todos os parâmetros sensíveis ao dano englobados no sistema de análise de vibrações proposto neste trabalho, sendo parametrizado de acordo com o exigido.

O sinal no tempo é reproduzido com recurso ao *Waveform Graph* (gráfico de forma de onda), que representa graficamente medições uniformemente amostradas. O espectro sinal no tempo não tem auto-escala, assim o utilizador pode adaptar a escala consoante os valores de amplitude obtidos. Caso o utilizador pretenda, o sinal no tempo originado é monitorizado a partir de alarmes máximos e mínimos, e executada através dos *express VIs Mask and Limit Testing* e *Amplitude and Level Measurements* visíveis no diagrama de blocos da Figura 4.34. Cada limite, máximo e mínimo, é estabelecido com recurso ao *express VI Amplitude and Level Measurements*. Por sua vez, o *Mask and Limit Testing* testa, por meio de funções de comparação, a saída do sinal de acordo com o limite mínimo ou máximo estabelecido em cada *express VI Amplitude and Level Measurements*.

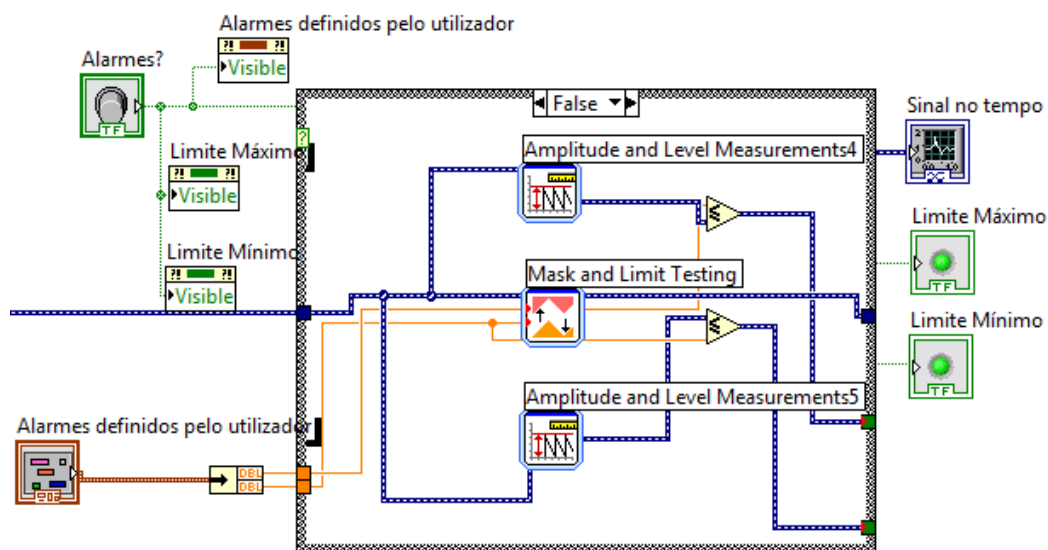


Figura 4.34: Diagrama de blocos do módulo sinal no tempo.

Os alarmes são exibidos na janela de interface com o utilizador em forma de linha horizontal colorida no espectro sinal no tempo e de indicadores luminosos do tipo booleano que se iluminam quando o limite é atingido. A cor verde corresponde ao limite máximo enquanto a azul corresponde ao limite mínimo. A título de exemplo a partir da Figura 4.31: se a amplitude máxima do sinal no tempo adquirido for de $0,8 \text{ m/s}^2$ (no caso de uma saída de sinal em velocidade), e se previamente o

utilizador tiver imposto como limite máximo $0,5 \text{ m/s}^2$), irá disparar o alarme correspondente no painel frontal avisando o utilizador.

4.5.3. Espectro FFT

A fim de fazer uma análise eficaz das características dos dados recolhidos na análise de vibrações real, é usada a FFT para passar do sinal no domínio do tempo para o sinal no domínio da frequência.

Dado que, o espectro FFT assume-se, no presente sistema de análise de vibrações, como o parâmetro sensível ao dano com mais opções e particularidades, este é também do ponto de vista programático mais complexo. De forma a simplificar alguns aspectos, a unidade de medida do espectro FFT é igual à escolhida para o sinal no tempo facilitando a navegação do utilizador e evitando alguma confusão por parte do mesmo.

O módulo espectro FFT permite que o utilizador escolha algumas propriedades, como é visível na Figura 4.35, por exemplo o tipo de amplitude de sinal, a escala do espectro, a janela para o atenuar o efeito de *leakage* e a possibilidade de utilizar médias (número de médias, ponderação e o tipo de média).

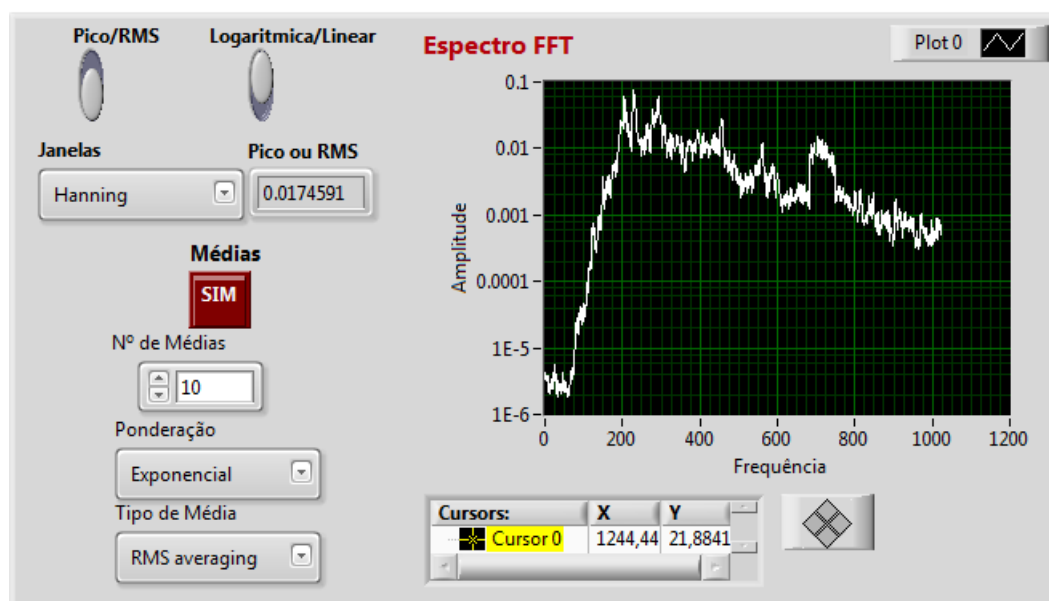


Figura 4.35: Painel frontal do módulo espectro FFT.

Sub-VI

Para ser possível visualizar o espectro FFT em pico ou em velocidade eficaz o espectro FFT é obtido com recurso à construção de um sub-VI personalizado denominado “pico e rms” que por sua vez é constituído por dois sub-VI semelhantes porém com objectivos diferentes. Em que, o sub-VI “pico” permite visualizar o espectro FFT com amplitude pico, enquanto o sub-VI “rms” permite visualizar o mesmo com amplitude RMS.

O sub-VI “pico” e o sub-VI “rms” são construídos da mesma forma, a partir do *express VI Spectral Measurements*. O referido *express VI* realiza medições espectrais baseadas no algoritmo FFT, tal como o espectro de magnitude média, o espectro de potência e o espectro de fase. Na janela de configuração do instrumento virtual *Spectral Measurements* escolhe-se a amplitude, pico ou RMS, o tipo de janela, a ponderação, o número de médias, entre outras características (ver Figura 4.36). Contudo, configurando o *express VI* no ambiente de computação, o utilizador, que apenas visualiza o painel frontal, não pode alterar estas características. Sabendo que, o objectivo do sistema de monitorização proposto é que este seja facilmente adaptável no futuro a outros processos industriais, o facto de o utilizador não ter a possibilidade de experimentar várias opções, não é proveitoso do ponto de vista prático.

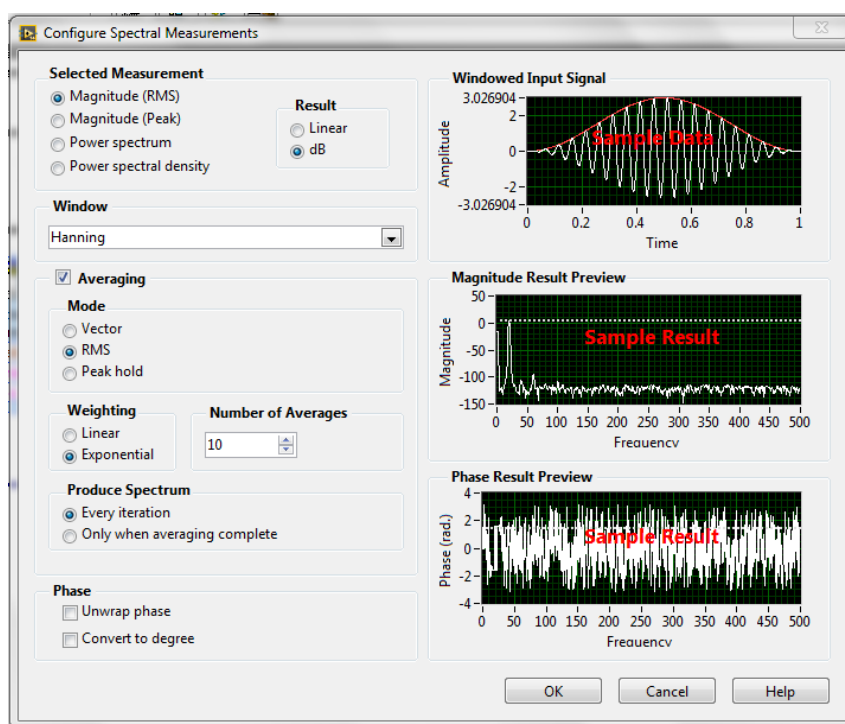


Figura 4.36: Configuração do *express VI Spectral Measurements* no diagrama de blocos.

Assim, é imprescindível alterar a lógica programática do *Spectral Measurements* para ir de encontro aos objectivos propostos. Para tal, e depois de se ter configurado o *Spectral Measurements* para: medir em RMS (*Magnitude (RMS)*), no caso da amplitude em RMS, e contemplar médias (*Averaging*); converte-se o *express VI* em sub-VI e após a conversão o sub-VI “rms”, está pronto a ser modificado.

A Figura 4.37 representa o diagrama de blocos do sub-VI “rms” antes das alterações feitas ao seu código, ou seja, tal como ele é projectado a partir da paleta de funções do LabVIEW. Observando o código verifica-se que as propriedades *Spectrum Window* (janelas do espectro) e *Spectrum Averaging Parameters* (parâmetros de médias do espectro) apresentam-se como constantes. O facto de serem constantes impossibilita, desde já a opção de escolha dessas propriedades por parte do utilizador no painel frontal, isto é, como se constatou anteriormente estas apenas podem ser personalizadas na janela de configuração do *express VI Spectral Measurements* (ver Figura 4.36).

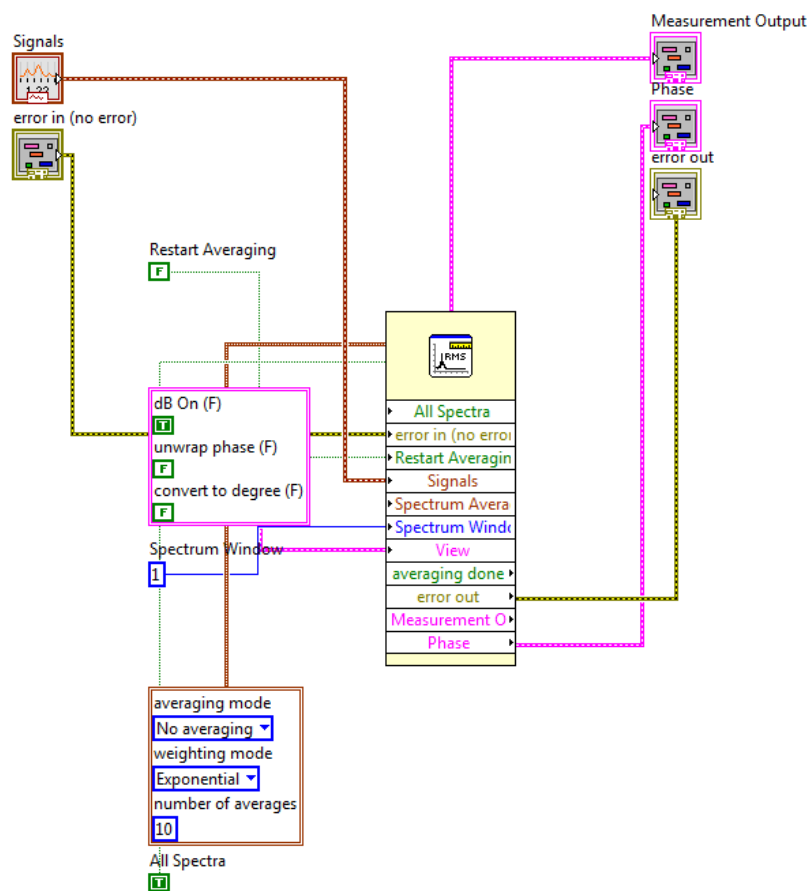


Figura 4.37: Diagrama de blocos do sub-VI “rms” pré-alterações.

Deste modo, a possibilidade de escolha do número de médias, tipo de ponderação, tipo de médias e tipo de janelas através de menus no painel frontal do espectro FFT (ver Figura 4.35) é realizada através de três etapas em que a primeira é a substituição das constantes, no diagrama de blocos do sub-VI “rms” (ver Figura 4.37), referentes às variáveis *Spectrum Averaging* e *Spectrum Window* por controladores numéricos do tipo *ring* (menu) visíveis na Figura 4.38. A Figura 4.38 exibe o diagrama de blocos do sub-VI “rms” após as transformações feitas ao código.

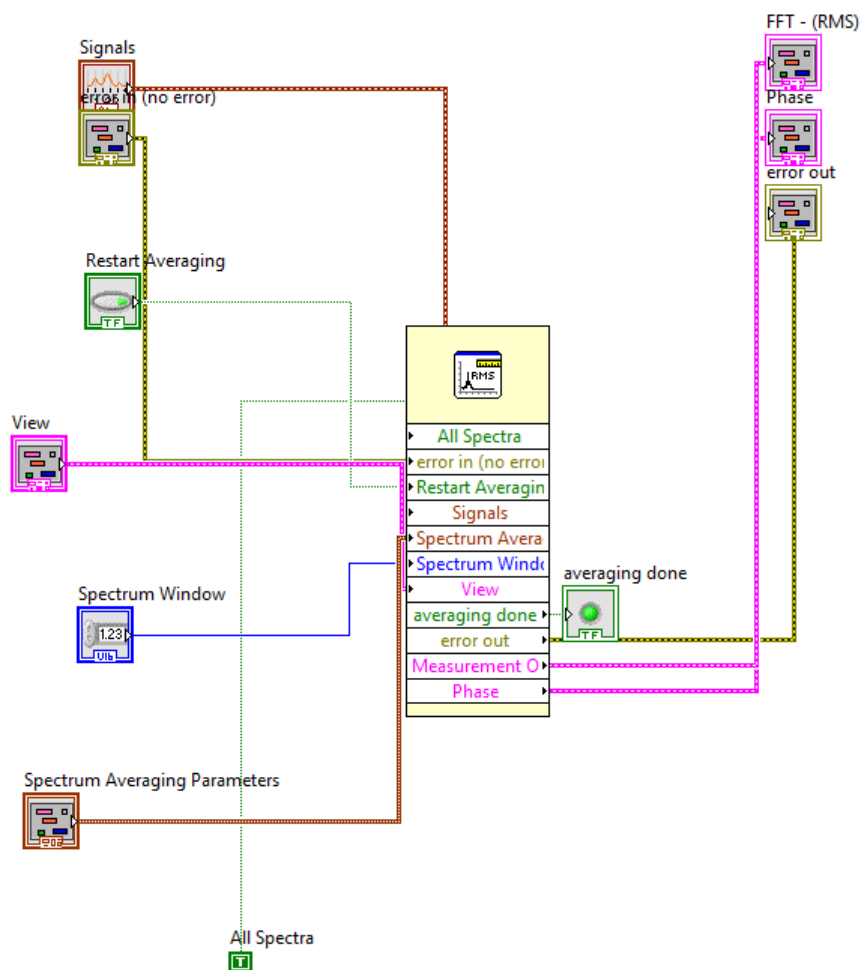


Figura 4.38: Diagrama de blocos do sub-VI “rms” pós-alterações.

Os controladores numéricos de *Spectrum Averaging Parameters* estão reunidos num *Cluster*. O *Cluster* é um grupo de elementos de dados que podem ser de diferentes origens (numérica, booleana, texto, etc), mas não agrupa no mesmo *Cluster* controladores e indicadores. Um *Cluster* é semelhante a um registo ou a uma estrutura, em linguagens de programação baseadas em texto. O *Cluster Spectrum Averaging Parameters* é do tipo valor numérico, este tipo de *Clusters* caracterizam-se por ter um fio de cor padrão castanho.

Para o sub-VI “pico” o procedimento é o mesmo, com a diferença de que na configuração do *express VI Spectral Measurements* a amplitude é em pico (*Magnitude (Peak)*).

Porém, como mencionado esta é apenas a primeira etapa para cumprir o objectivo da possibilidade de escolha do número de médias, tipo de ponderação, tipo de médias e tipo de janelas através de menus no painel frontal do espectro FFT (ver Figura 4.35).

De seguida, na segunda etapa, resta construir o sub-VI “pico e rms” (ver Figura 4.39) cujo objectivo é poder visualizar o espectro FFT em pico ou em RMS. Para satisfazer esse objectivo é preciso conectar os dados relativos a cada sub-VI começando pelos sinais de saída FFT do sub-VI “rms” e do sub-VI “pico” a uma função de comparação do tipo *Select* (selecção), que retorna o valor ligado à entrada “*t*” (verdadeira) ou à entrada “*f*” (falsa), dependendo do valor de “?”. Se “?” for verdadeira, a função retorna o valor ligado a “*t*”. Se “?” for falsa, a função retorna o valor ligado a “*f*”. Neste caso, “?” é um controlador do tipo booleano “pico/rms”, “*t*” é o sinal FFT em pico (FFT – (*Peak*)) e “*f*” o sinal FFT em RMS (FFT – (RMS)). O resultado desta função é o sinal de FFT escolhido no painel frontal pelo utilizador (pico ou RMS) e a visualização do mesmo através de um espectro tornado possível a partir do VI *Waveform*.

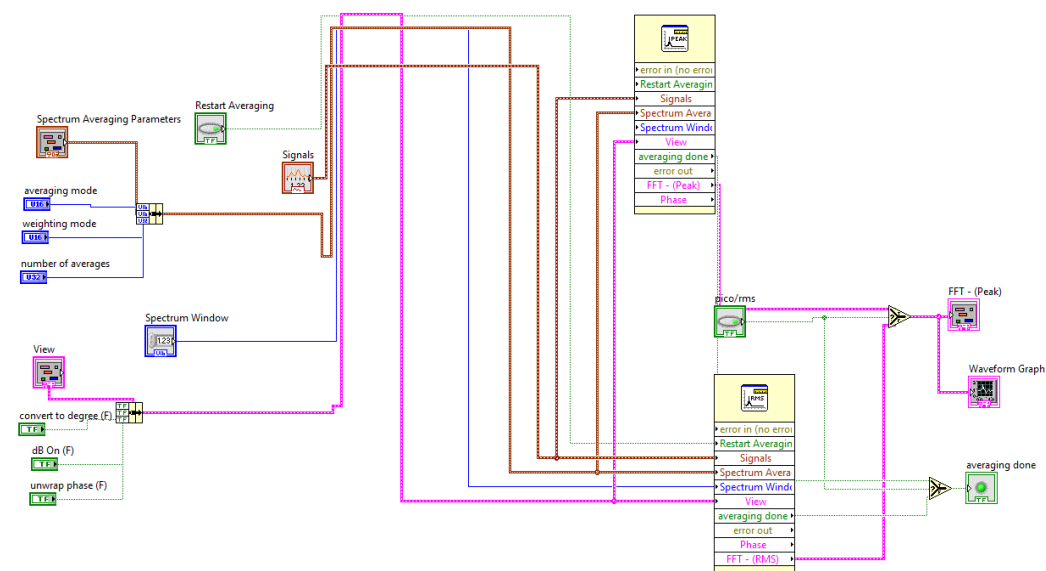


Figura 4.39: Diagrama de blocos do sub-VI “pico e rms”.

Voltando à possibilidade de inclusão de médias no espectro FFT, os parâmetros das médias do espectro (número de médias, tipo de ponderação e tipo de média)

estão reunidos num *Cluster* de dados *Spectral Averaging Parameters* mas, para ser executável alterar os valores individuais de cada elemento sem ter que especificar novos valores para cada é imprescindível utilizar uma função *Bundle* (ver Figura 4.40). Uma *Bundle* reúne um *Cluster* a partir de elementos individuais. Para fazê-lo conecta-se o *Cluster* dos parâmetros das médias do espectro ao terminal de *Cluster* no meio do *Bundle*, a partir daqui a função redimensiona automaticamente exibindo as entradas para cada elemento constituinte do *Cluster*: *averaging mode* (tipo de média), *weighting mode* (tipo de ponderação) e *number of averages* (número de médias). O *Bundle* do *Cluster* vai permitir o utilizador seleccionar a opção pretendida em cada menu no painel frontal do espectro FFT (ver Figura 4.35).

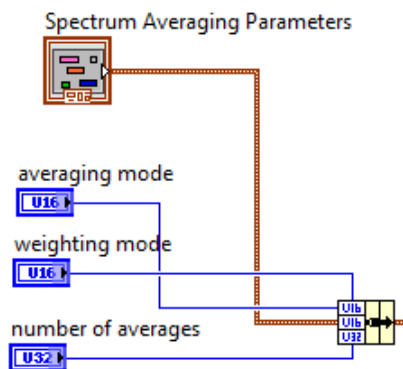


Figura 4.40: *Bundle* de elementos no sub-VI “pico e rms”.

A terceira e derradeira etapa para a construção do módulo espectro FFT com todas as suas características é a inserção do sub-VI “pico e rms” numa *Case*, exibida na Figura 4.41, com dois casos controlados pelo botão de decisão “Médias”. O caso verdadeiro é a inclusão de médias no espectro FFT e o caso falso a exclusão de médias no espectro FFT. As *rings* inerentes aos parâmetros de médias do espectro e das janelas são conectadas às entradas correspondentes no sub-VI “pico e rms” e são visíveis no painel frontal por parte do utilizador (ver Figura 4.35). O utilizador pode ir verificando o valor de pico ou RMS conforme o pretendido, através de um indicador numérico (“Pico ou RMS”). Esta propriedade é obtida com recurso ao *express VI Amplitude and Level Measurements* que mede o valor de pico e de RMS, e à função *Select* que retorna o valor da entrada pico (“*t*”) ou da entrada RMS (“*f*”), consoante o resultado do botão de decisão “Pico/RMS” (“?”).

A escolha da escala de amplitudes, linear ou logarítmica, é independente do tipo de amplitude escolhida (pico ou RMS). Esta opção é possível através da inserção de um *property node* do tipo *mapping mode* no eixo da amplitude (*Yscale*), que especifica o modo de mapeamento da escala. O *mapping mode* é controlado através de um controlo booleano “Logarítmica/Linear”, visível no painel frontal do espectro FFT (ver Figura 4.35) e da função de comparação *Select*, que retorna o valor ligado à entrada “*t*” ou à entrada “*f*”, dependendo do valor de “?”. Aqui, “?” é o botão de decisão no painel frontal, “*t*” é igual a 1 (escala logarítmica) e “*f*” igual a 0 (escala linear).

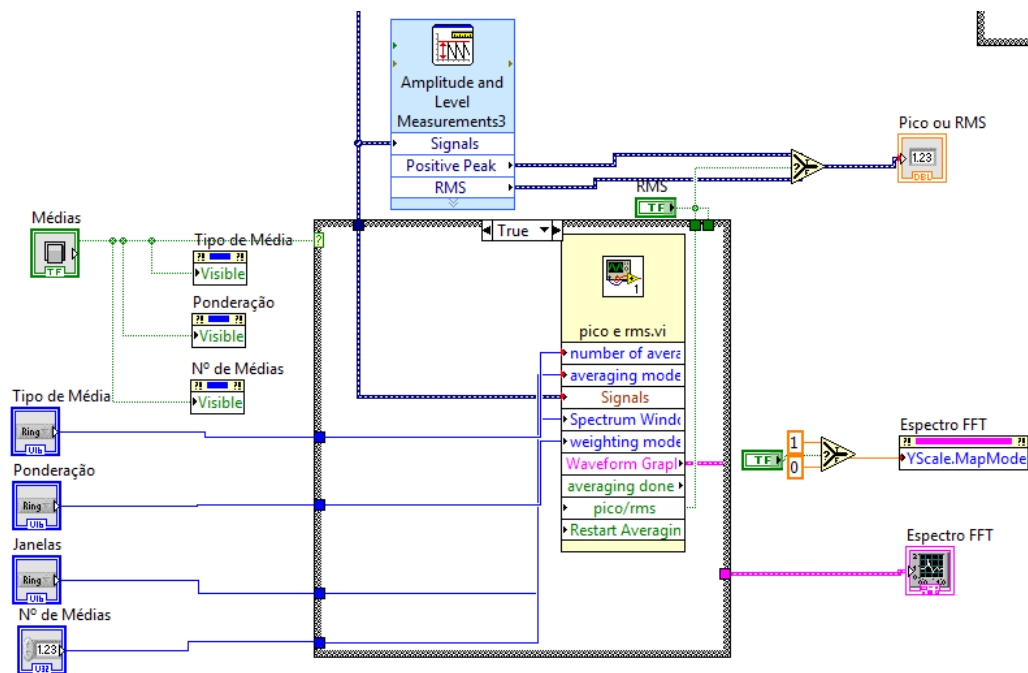


Figura 4.41: Diagrama de blocos do módulo espectro FFT.

4.5.4. Factor de Crista

O factor de crista é determinado através do sinal resultante da estrutura *Case* “Velocidade” (ver Figura 4.32), isto é, o sinal em velocidade. Para qualquer tipo de sensor de entrada: deslocamento, velocidade ou aceleração, o sinal irá sair sempre em velocidade através de integrações e derivações. Dado que, o factor de crista é especialmente adequado para caracterizar sinais que contêm impulsos repetitivos, como a vibração produzida por falhas em rolamentos, no sistema de análise de vibrações proposto o factor de crista é medido em aceleração, filtrada de 1000 a 10000 Hz. Para tal, o sinal é derivado através de um função de derivação obtendo-se o sinal pretendido em aceleração e após a derivada é aplicado um filtro

do tipo passa-banda (*Bandpass*) que condiciona o sinal entre 1000 e 10000 Hz (ver Figura 4.42).

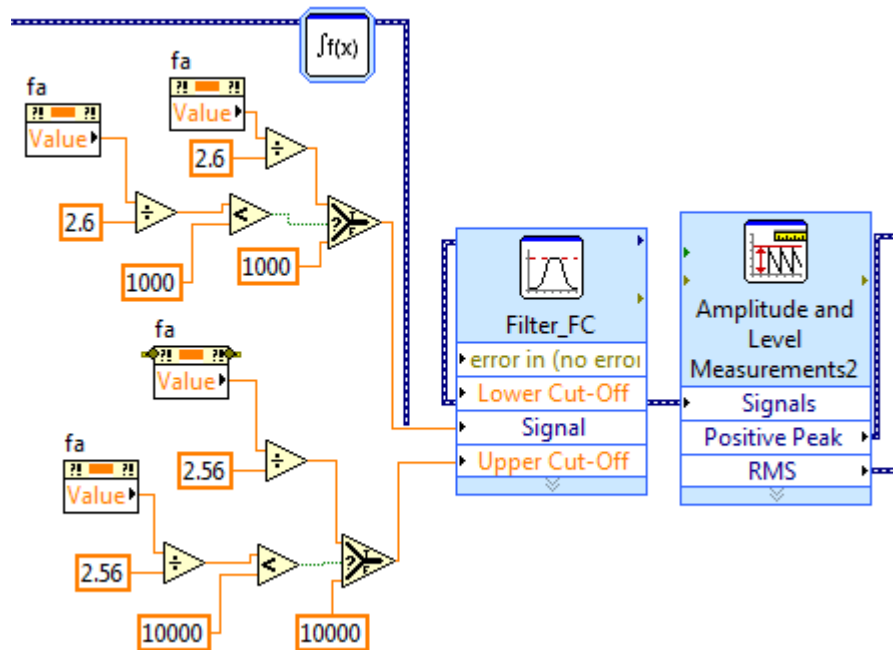


Figura 4.42: Configuração do filtro anti-aliasing aplicado ao sinal do factor de crista.

Como o filtro passa-banda não evita a atenuação de algumas frequências um pouco antes da frequência limite devido ao fenómeno de *aliasing*, é vital providenciar um filtro anti-aliasing. O filtro anti-aliasing é construído pela criação de um *lower cut-off* (frequência de corte inferior) e um *upper cut-off* (frequência de corte superior) com a ajuda de *express VIs* de comparação e de selecção (ver Figura 4.42). Se f_a a dividir por 2,56 for inferior a 10000 Hz o *upper cut-off* assume o valor de f_a a dividir por 2,56. Por outro lado, se o terminal booleano “?” for falso então o *upper cut-off* vai assumir o valor de 10000 Hz, garantindo sempre que irá filtrar entre 1000 e 10000 Hz.

De seguida, o sinal é processado através do instrumento virtual *Amplitude and Level Measurements*, visível na Figura 4.42, que determina o valor de pico e o valor de RMS. O quociente entre o pico e o RMS estabelece o valor do factor de crista que é exibido no painel de interface com o utilizador a partir de um indicador numérico (ver Figura 4.43).

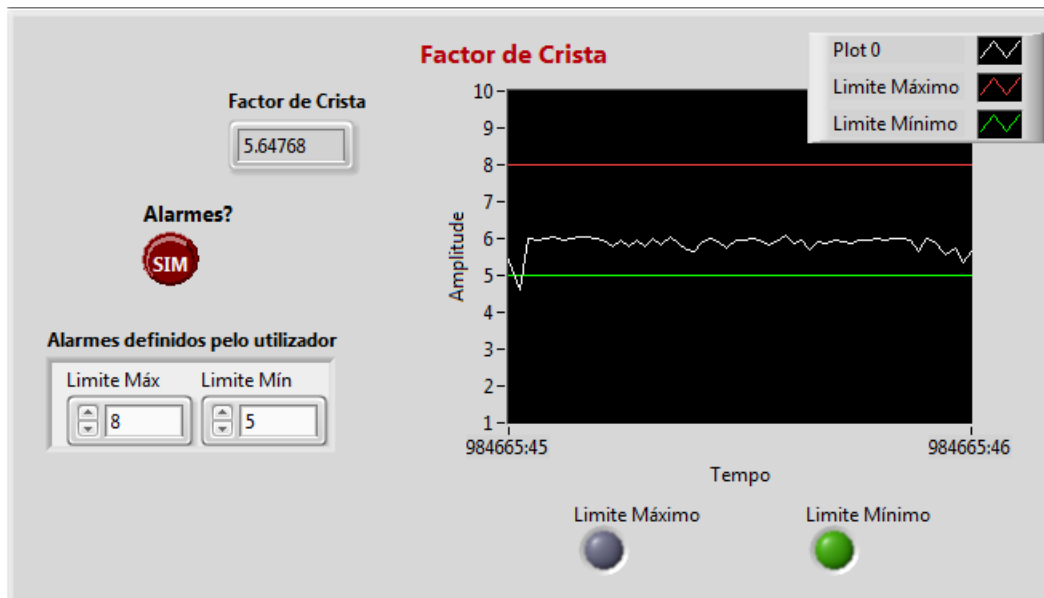


Figura 4.43: Painel frontal do módulo factor de crista.

No diagrama de blocos, o sinal prossegue para a *Case* responsável pela produção do gráfico factor de crista, que apresenta a variação do factor de crista ao longo do tempo (ver Figura 4.44).

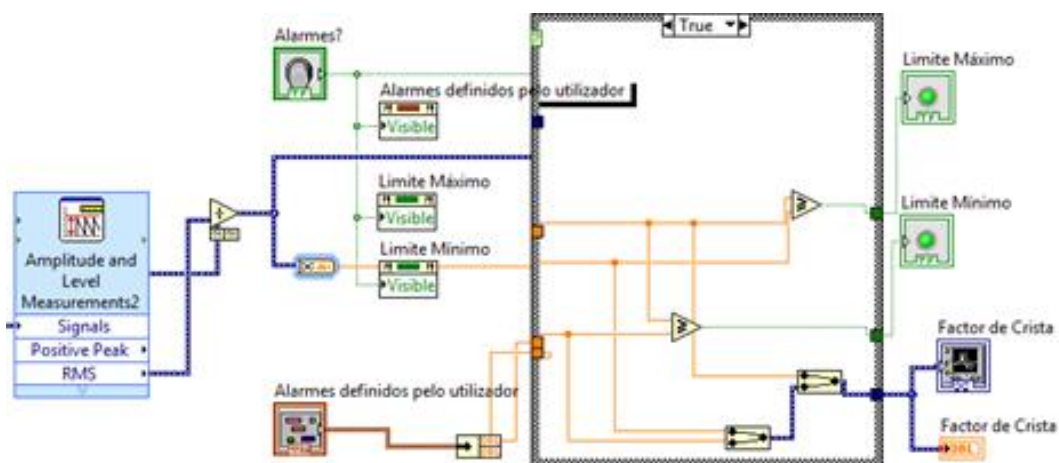


Figura 4.44: Diagrama de blocos do módulo factor de crista.

De forma análoga ao sinal no tempo, nesta janela é dada a hipótese ao utilizador de definir valores limite de alarme. Os valores limite são comparados com o sinal adquirido, através de funções de comparação visíveis na *Case* do diagrama de blocos.

4.5.5. Nível Global

O painel frontal do módulo nível global é constituído pela representação indicativa do valor do nível global, através de um indicador numérico e pelo gráfico que exhibe

o valor do mesmo ao longo do tempo com os respectivos sinais de alarme (ver Figura 4.45).

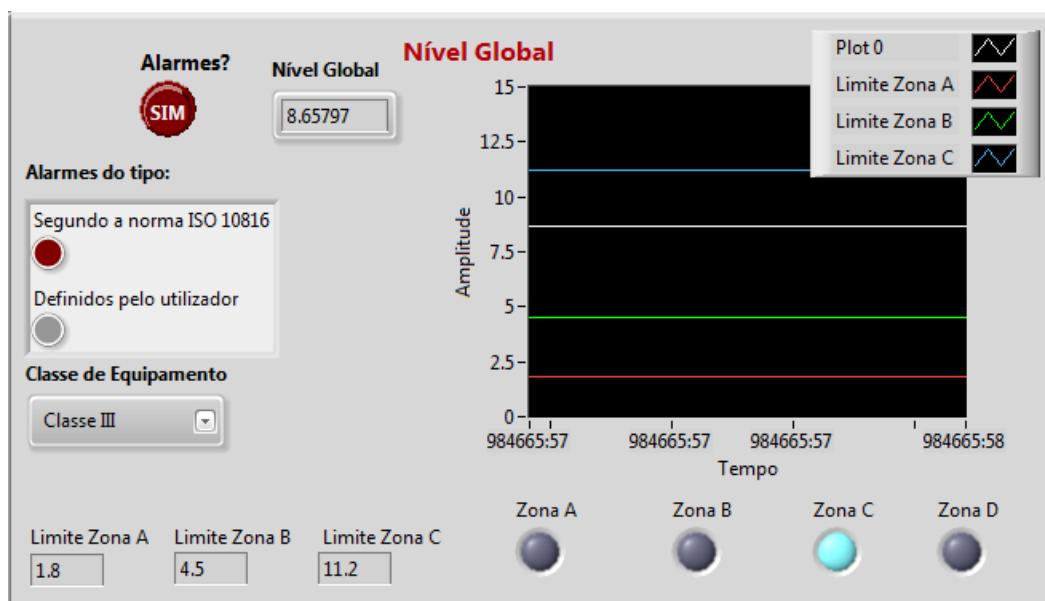


Figura 4.45: Painel frontal do módulo nível global.

O utilizador escolhe que tipo de alarmes quer experimentar: alarmes definidos por ele próprio (limite mínimo e limite máximo) ou alarmes segundo a comparação dos valores medidos com tabelas de severidade de vibração. Optou-se do ponto de vista didático, por fazer a comparação é feita com os critérios de severidade da norma ISO 10816-1. A parte 1 da norma ISO 10816 estabelece orientações gerais para medição e avaliação de vibrações mecânicas de máquinas medidas em componentes não-rotativos. A tabela de severidade de vibração para valores RMS em milímetros por segundo (ver Tabela 4.4) é dividida por classes:

- Classe I – “Componentes individuais de motores e máquinas (até 15kW), integralmente conectadas à máquina completa e na sua condição de operação nominal.”;
- Classe II – “Máquinas de médio porte sem fundações especiais (de 15kW até 75kW), motores montados rigidamente ou máquinas com fundação especiais (até 300kW).”;
- Classe II – “Máquinas de grande porte com partes rotativas montadas em fundações rígidas e pesadas que são relativamente rígidas na direcção da medição de vibrações.”;

- Classe IV – “Máquinas de grande porte com partes rotativas montadas em fundações que são relativamente macias na direcção da medição de vibrações.”.

E por zonas:

- Zona A – Bom. Vibrações de máquinas novas durante a sua recepção;
- Zona B – Satisfatório. As máquinas podem ser utilizadas sem restrições;
- Zona C – Insatisfatório. As máquinas podem ser utilizadas somente durante um período de tempo limitado;
- Zona D – Inaceitável. As vibrações são perigosas podendo ter consequências negativas para as máquinas.

Tabela 4.4: Tabela de severidade da norma ISO 10816-1

RMS [mm/s]	Class I	Class II	Class III	Class IV	
0,28	A	A	A	A	
0,45					
0,71					
1,12	B	B	B	B	
1,8					
2,8	C	C	B	B	
4,5					
7,1	D	D	C	C	
11,2					
18			D	D	D
28					
45					

O utilizador tem a oportunidade de escolher e adequar a classe ao equipamento do qual está a adquirir sinal e verificar em qua zona se encontra o mesmo através da indicação luminosa de alarmes para cada zona. Os valores limite das zonas correspondentes à classe seleccionada também são visíveis no painel de interface com utilizador através de indicadores numéricos que se alteram de acordo com a classe (Figura 4.45).

No ambiente de computação o nível global é calculado através do sinal que sai da *Case* do tipo de sensor, tal como no factor de crista este sinal sai sempre em velocidade, independentemente do tipo de sensor escolhido no módulo de configuração dos dados de aquisição (ver diagrama de blocos do sistema de análise de vibrações no Anexo B, página 125). O sinal é filtrado em 10 e 1000 Hz e é previsto o filtro anti-*aliasing* de forma idêntica à descrita no factor de crista. No *express VI Amplitude and Level Measurements* a amplitude do sinal medida é o RMS. Os dados dinâmicos são processados em três estruturas *Case* em que a primeira corresponde à decisão do utilizador de introduzir alarmes e é controlada por um controlo booleano. A segunda *Case* corresponde à decisão por parte do utilizador em relação ao tipo de alarmes que quer considerar: alarmes definidos por si ou alarmes segundo a norma ISO 10816, e onde se encontram as funções de comparação para a verificação entre os valores de nível global medidos e os valores limite (os ISO 10816 ou os definidos pelo utilizador). A terceira e última *Case* possibilita:

- ✓ visualizar individualmente no painel frontal os indicadores numéricos referentes aos limites das zonas para cada classe de equipamento,
- ✓ ajudar na comparação dos valores medidos de nível global com os valores da tabela de severidade da ISO 10816,
- ✓ e visualizar no gráfico do nível global os alarmes em forma de linha colorida indicativa através da função *Merge Signals* do LabVIEW que permite unir dois ou mais sinais, escalares e/ou dinâmicos.

5. Conclusões

O capítulo 5. expõe, na secção 5.1, as conclusões gerais deste Trabalho Final de Mestrado, analisando a aplicabilidade do sistema de detecção e diagnóstico de falhas via Internet proposto. Na secção 5.2 indicam-se algumas melhorias exequíveis como trabalho futuro que podem ser implementadas ao sistema proposto.

5.1. Trabalho Final

O presente Trabalho Final de Mestrado tinha como principal objectivo a criação de um sistema de diagnóstico de falhas via Internet de processos industriais controlados por PLCs e pode afirmar-se que este objectivo foi cumprido. Todavia, foram necessárias várias etapas para atingi-lo.

Primeiramente, procedeu-se ao levantamento das variáveis a monitorizar e constatou-se que nos processos industriais a maioria das variáveis são lógicas. Além disso, os recursos existentes no laboratório de Automação Industrial da Área Departamental de Engenharia Mecânica, ISEL, não permitiram por exemplo a implementação de um sistema de elementos redundantes (sensores, motores, entre outros). Por estas razões, a análise e a procura de conhecimento entre inúmeros métodos de detecção e diagnóstico de falhas revelou-se uma tarefa complexa e morosa que culminou na escolha dos métodos utilizados.

Deste modo, desenvolveu-se um sistema que tem por base a monitorização, detecção e o diagnóstico das variáveis de entrada e saída dos elementos testados nas experiências existentes no LabAPI – ISEL, através do método de verificação de limites fixos e de métodos de diagnóstico, nomeadamente a partir da aplicação de árvores de falha com lógica booleana e de regras IF-THEN. O sistema foi criado a partir do programa de linguagem gráfica LabVIEW e a interface cliente/servidor foi estabelecida através do protocolo OPC. A página *web* do sistema de detecção e diagnóstico de falhas proposto pode ser acedida por utilizadores internos ou externos, com a possibilidade de manipulação total do painel frontal por parte dos mesmos, conforme as suas permissões.

Tendo em mente que a análise de vibrações tem também um papel preponderante no campo da supervisão de máquinas ou de monitorização de um processo industrial, complementarmente propôs-se um sistema de análise de vibrações.

A principal dificuldade da implementação da página *web* do sistema FDI proposto prendeu-se essencialmente com a decisão de quais as falhas possíveis de detectar e diagnosticar, tendo em consideração as variáveis lógicas monitorizadas. Outra das dificuldades enfrentadas foi a aprendizagem e aprofundamento do conhecimento de programação através de linguagem gráfica.

O trabalho atrás descrito servirá também para dar apoio às aulas das unidades curriculares de Programação de Autómatos e Automação de Processos Industriais.

5.2. Trabalho Futuro

O sistema de detecção e diagnóstico de falhas via Internet de processos industriais proposto e testado no LabAPI – ISEL tem algumas limitações que devem vir a ser melhoradas no futuro. As principais melhorias a executar são as seguintes:

- ✓ Tornar o sistema FDI acessível a partir do *browser* Google Chrome.
- ✓ Incluir a monitorização, detecção e o diagnóstico de falhas de rede no sistema FDI implementado. Respeitando os atrasos na rede para garantir que todos os equipamentos industriais (PLCs, placa de aquisição de dados, entre outros) recebam a informação remota numa janela temporal predefinida pelas limitações da aplicação.
- ✓ Implementar uma base de dados do sistema FDI com o armazenamento dos dados adquiridos e criação de um histórico, que possa ser consultado em tempo-real pelos utilizadores. Esta base de dados pode ser obtida com recurso ao módulo Datalogging and Supervisory Control (DSC) do programa LabVIEW. O DSC permite registar dados (relatórios, históricos, entre outros) numa base de dados – Citadel. A Citadel faz parte integral de muitos produtos National Instruments e também permite conectar o sistema desenvolvido a uma base de dados já existente. Por exemplo, integrar a base de dados do sistema FDI com a base de dados do *website* LabAPI – ISEL (executada em MySQL).

- ✓ Aperfeiçoar a interface com o utilizador colocando as imagens das experiências “Pórtico” e “Cilindros” em movimento de acordo com a execução das experiências.
- ✓ Melhorar o sistema de análise de vibrações, a partir da inclusão de acelerómetros nos três motores que o pórtico possui de forma a obter dados de aquisição reais.
- ✓ Melhorar o sistema de gestão de acessos, através do desenvolvimento de um sistema de agendamento das experiências, permitindo aos utilizadores do laboratório remoto reservarem a experiência desejada para o dia e hora pretendido.

Por último, referir que o sistema de detecção e diagnóstico de falhas proposto é apenas um exemplo do que pode ser aplicado na indústria, podendo ser utilizados outro tipo de sistemas com variáveis lógicas diferentes com recurso a outro tipo de técnicas FDI.

Referências Bibliográficas

- [1] M. Roth, S. Schneider, J.-J. Lesage e L. Litz, “Fault detection and isolation in manufacturing systems with an identified discrete event model,” *International Journal of Systems Science*, vol. 43, pp. 1826-1841, 2012.
- [2] R. Sekar, S.-J. Hsieh e Z. Wu, “Remote Diagnosis Design for a PLC-based automated system: 1 - Implementation of Three Levels of Architectures,” *Springler-Verlag London Limited*, 2011.
- [3] I. Michaelides, P. Eleftheriou e K. Economides, “Remote engineering education: real-world experiments in solar energy over the internet,” *MARVEL - A Leonardo da Vinci Pilot Project*, p. 39, 2005.
- [4] M. Gurney, “System integration enables Industrial Internet of Things, Industrie 4.0 platforms,” *Control Engineering*, pp. 35-36, Junho 2016.
- [5] S. A. A. Amin, A. Ali-Eldin e H. A. Ali, “A context-aware dispatcher for the Internet of Things: The case of electric power distribution systems,” *Computers and Electrical Engineering*, vol. 52, pp. 183-198, Junho 2015.
- [6] P. Thamarai e R. Amudhevali, “Energy Monitoring System Using PLC & SCADAS,” *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*, vol. 3, pp. 7126-7133, 2014.
- [7] C. Salzmann e D. Gillet, “Challenges in Remote Laboratory Sustainability,” em *International Conference on Engineering Education*, Coimbra, 2007.
- [8] L. M. S. Martins, “Projecto dum Laboratório Remoto para Automação de Processos Industriais,” ISEL, Lisboa, 2013.

- [9] R. M. F. Matias, “Monitorização via Internet de Processos Industriais Controlados com Autómatos Programáveis,” ISEL, Lisboa, 2014.
- [10] R. Isermann, *Fault-diagnosis systems: an introduction from fault detection to fault tolerance*, Berlin: Springer, 2006.
- [11] M. Albert, “Seven Things to Know About the Internet of Things and Industry 4.0,” *Modern Machine Shop*, pp. 74-81, Setembro 2015.
- [12] W. MacDougall, “Germany Trade & Invest,” Julho 2014. [Online]. Available:
http://www.gtai.de/GTAI/Content/EN/Invest/_SharedDocs/Downloads/GTAI/Brochures/Industries/industrie4.0-smart-manufacturing-for-the-future-en.pdf. [Acedido em 10 Agosto 2016].
- [13] VDMA, “Industry 4.0: The Next Industrial Revolution to Take Place in Germany,” *VDMA Mining Supplement*, pp. 14-15, 2015.
- [14] R. Neuhauser, “Shaping the Future of Production with Siemens and Sinumerik,” em *EMO*, Hannover, Alemanha, 2013.
- [15] R. Harrison, D. Vera e B. Ahmad, “Engineering Methods and Tools for Cyber-Physical Automation Systems,” *Proceedings of the IEEE*, vol. 104, pp. 973-985, Maio 2016.
- [16] P. Leitão, A. W. Colombo e S. Karnouskos, “Industrial automation based on cyber-physical systems technologies: Prototype implementations and challenges,” *Computers in Industry*, vol. 81, pp. 11-25, Setembro 2015.
- [17] D. Zhang, J. Wan, C.-H. (. Hsu e A. Rayes, “Industrial technologies and applications for the Internet of Things,” *Computer Networks*, vol. 101, p. 1, 2016.

- [18] R. Kirubashankar, K. Krishnamurthy, J. Indra e B. Vignesh, “Design and Implementation of Web Based Remote Supervisory Control and Information System,” *International Journal of Soft Computing and Engineering (IJSCE)*, vol. I, pp. 43-51, 2011.
- [19] Schneider, “Schneider Electric,” Março 2012. [Online]. Available: http://www.schneider-electric.com/solutions/ww/fr/med/20340568/application/pdf/1485_se-whitepaper-letter-scadaoverview-v005.pdf. [Acedido em 10 Dezembro 2014].
- [20] A. M. L. G. Borracha, “Laboratório Remoto de Automação Industrial (Lab-RAI),” Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa, Lisboa, 2012.
- [21] K. Zhang, “Fault Detection and Diagnosis for A Multi-Actuator Pneumatic System,” Stony Brook University, Nova Iorque, 2011.
- [22] R. Isermann e P. Ballé, “Trends in the application of model-based fault detection and diagnosis of technical processes,” Elsevier Science Ltd, Darmstadt, 1997.
- [23] R. Sekar, “Analysis of Remote Diagnosis Architecture for a PLC Based Automated Assembly System,” Texas, EUA, 2010.
- [24] C. F. A. Calmeiro, *Tolerância a falhas com base em comutação de controladores - Implementação em autómatos programáveis*, Caparica: Faculdade de Ciências e Tecnologia - Universidade Nova de Lisboa, 2013.
- [25] M. Blanke, M. Kinnaert, J. Lunze e M. Staroswiecki, “Faults and fault tolerance,” em *Diagnosis and Fault-Tolerant Control*, Alemanha, Springer, 2006, pp. 6-8.

- [26] S. D. d. S. Almeida, “Controlador Tolerante a Falhas para Veículo Eléctrico Multi-Motor,” Faculdade de Engenharia da Universidade do Porto, Porto, 2013.
- [27] R. Isermann, “Supervision, fault-detection and fault-diagnosis methods - an introduction,” Elsevier Science Ltd, Darmstadt, 1997.
- [28] F. Feather, D. Slewioerek e R. Maxion, “Fault detection in an Ethernet network using anomaly signature matching,” em *SIGCOMM'93-Ithaca*, EUA, 1993.
- [29] Y. Pencolé, “Diagnosti décentralisé de systèmes à événements discrets: Appication aux réseaux de télécommunication,” França, 2002.
- [30] B. Brahimi, C. Aubrun e E. Rondeau, “Network Calculus Based FDI Approach For Switched Ethernet Architecture,” em *Fault Detection, Supervision and Safety of Technical Processes 2006*, 1ª ed., China, Elsevier Science, 2007, pp. 312-313.
- [31] A. K. Jardine, D. Lin e D. Banjevic, “A review on machinery diagnostics and prognostics implementing condition-based maintenance,” *Mechanical Systems and Signal Processing*, pp. 1484-1498, 2005.
- [32] L. Ribeiro e J. Barata, “Re-thinking diagnosis for future automation systems: An analysis of current diagnostic practices and their applicability in emerging IT based production paradigms,” *Computers in Industry*, vol. 62, pp. 639-659, 2011.
- [33] J. J. Gertler, “Introduction to Fault Detection and Diagnosis,” em *Fault Detection and Diagnosis in Engineering Systems*, Basel, Marcel Dekker, Inc., 1998, pp. 2-8.

- [34] R. Isermann, "Model-based fault detection and diagnosis - status and applications," Institute of Automatic Control, Darmstadt University of Technology, Darmstadt, 2004.
- [35] R. Isermann, "Fault diagnosis of machines via parameter estimation and knowledge processing - Tutorial paper," *Automatica*, vol. 29, pp. 815-835, 1992.
- [36] P. M. Frank, "Fault diagnosis in dynamic systems using analytical and knowledge-based redundancy: A survey and some new results," *Automatica*, vol. 26, pp. 459-474, 1990.
- [37] J. Gertler, "Fault detection and isolation using parity relations," *Control Engineering Practice*, vol. 5, pp. 653-661, 1998.
- [38] R. Isermann, D. Füssel e H. Straky, "Industrial applications of fault diagnosis," Encyclopedia of Life Support Systems (EOLSS), Darmstadt.
- [39] V. Venkatasubramanian, R. Rengaswamy e S. N. Kavuri, "A review of process fault detection and diagnosis Part II: Qualitative models and search strategies," Elsevier Science Ltd., Houston, 2002.
- [40] S. S. Rao, "Fundamentals of Vibration," em *Mechanical Vibrations*, 5^a ed., EUA, Prentice Hall, 2011, pp. 10-63.
- [41] I. Lita, D. A. Visan, G. Mujea e D. Ghita, "LabVIEW Application for Analysis of Mechanical Vibrations from Industrial Environment," em *28th Int. Spring Seminar on Electronics Technology*, 2005.
- [42] Z. Hua, "Application of LabVIEW in the Design of Data Acquisition and Signal Processing System of Mechanical Vibration," em *2011 International Conference on Mechatronic Science, Electric Engineering and Computer*, Jilin, China, 2011.

- [43] J. A. Stamm, "Machinery Diagnostics Via Mechanical Vibration Analysis Using Spectral Analysis Techniques," Naval Postgraduate School, Califórnia, EUA, 1988.
- [44] L. R. Higgins, R. K. Mobley e D. Wikoff, "Vibration: Its Analysis and Correction," em *Maintenance Engineering Handbook*, McGraw-Hill, 2014.
- [45] C. M. Harris, "Introduction to the Handbook," em *Harris' Shock and Vibration Handbook*, 5ª ed., EUA, McGraw-Hill, 2002, pp. 1.1-1.3; 1.20.
- [46] J. T. Broch, "Vibration Measuring Instrumentation and Techniques," em *Mechanical Vibration and Shock Measurements*, 2nd ed., Denmark, Brüel & Kjaer, 1984, pp. 97-114.
- [47] S. S. Rao, "Vibration Measurement and Applications," em *Mechanical Vibrations*, 5ª ed., EUA, Prentice Hall, 2011, pp. 895-924.
- [48] D. H. Shreve, "Signal Processing For Effective Vibration Analysis," 1995.
- [49] V. Instruments, "VTI Instruments," [Online]. Available: www.vtiinstruments.com/CATALOG/Technotes/Technote12.pdf. [Acedido em 28 Fevereiro 2016].
- [50] R. B. Randall, "Vibration Analyzers and Their Use," em *Harris' Shock And Vibration Handbook*, 5ª ed., EUA, McGraw-Hill, 2002, pp. 14.13-14.17; 14.29.
- [51] R. B. Randall, "Vibration Measurement Instrumentation," em *Harris' Shock and Vibration Handbook*, 5ª ed., EUA, McGraw-Hill, 2002, pp. 13.5-13.10.
- [52] S. S. Rao, "Vibration Control," em *Mechanical Vibrations*, 5ª ed., EUA, Prentice Hall, 2011, pp. 771-773.

- [53] A. Friedman, “AzimaDLI,” [Online]. Available: <http://azimadli.com/wp-content/uploads/TECH-CrestFactor.pdf>. [Acedido em 20 Março 2016].
- [54] J. Courrech e R. L. Eshleman, “Condition Monitoring of Machinery,” em *Harris' Shock and Vibration Handbook*, 5ª ed., EUA, McGraw-Hill, 2002, pp. 16.17-16.21.
- [55] S. Da'na, A. Sagahyoon, A. Elrayes, A. Al-Ali e R. Al-Aydi, “Development of a monitoring and control platform for PLC-based applications,” *Computer Standards & Interfaces*, vol. 30, pp. 157-166, Agosto 2007.
- [56] L. F. F. B. Palma, “Fault Detection, Diagnosis and Fault Tolerance Approaches in Dynamic Systems Based on Black-box Models,” Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa, Lisboa, 2007.
- [57] M. J. G. C. Mendes, “Multi-agent Approach to Fault Tolerant Control Systems,” Instituto Superior Técnico, Universidade Técnica de Lisboa, Lisboa, 2008.
- [58] L. Catani, “Extending LabVIEW Aptitude for Distributed Controls and Data Acquisition,” Instituto Nacional de Física Nuclear, Universidade de Roma Tor Vergata, Itália, 2011.
- [59] H. Vargas, J. Sánchez-Moreno, S. Dormido, C. Salzmann, D. Gillet e F. Esquembre, “Web-Enabled Remote Scientific Environments,” *Computing in Science & Engineering*, p. 36, 2009.
- [60] N. Instruments, “3. Tipos de dados e estruturas,” [Online]. Available: <http://www.ni.com/f/students/12/7068/pt>. [Acedido em 15 Fevereiro 2016].

- [61] N. Instruments, “Introduction to OPC,” 7 Setembro 2012. [Online]. Available: <http://www.ni.com/white-paper/7451/en/>. [Acedido em 11 Fevereiro 2016].
- [62] N. Instruments, “Módulo LabVIEW Datalogging and Supervisory Control (DSC),” [Online]. Available: <http://www.ni.com/labview/labviewdsc/pt/>. [Acedido em 10 Julho 2016].
- [63] N. Instruments, “Manuais de produtos NI,” [Online]. Available: <http://www.ni.com/pdf/manuals/371303n.pdf>. [Acedido em 29 Julho 2015].
- [64] W3Schools, “Browsers Statistics,” [Online]. Available: <http://www.w3schools.com/browsers/default.asp>. [Acedido em 20 Setembro 2016].

Anexo A

Menu da Experiência “Cilindros”:



Laboratório Remoto
de Automação de
Processos Industriais

Experiência: Cilindros

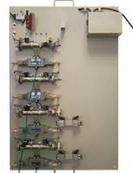
[SAIR](#)



[Ambiente de Trabalho Remoto](#)



[Web Server S7-1200](#)



[Webcam](#)



[Sistema de Detecção e Diagnóstico de Falhas](#)

Menu da Experiência “Pórtico”:



Laboratório Remoto
de Automação de
Processos Industriais

Experiência: Pórtico

[SAIR](#)



[Ambiente de Trabalho Remoto](#)



[Web Server S7-1200](#)



[Webcam](#)



[Sistema de Detecção e Diagnóstico de Falhas](#)

Sistema de Detecção e Diagnóstico de Falhas Via Internet:

Sistema de Detecção e Diagnóstico de Falhas

Por favor faça login de acordo com as suas permissões.

The screenshot shows a web browser window with the following elements:

- Header:** "Analisador de Vibrações" and "Controles On/Off".
- Logos:** ISEL and ADEM (Área Departamental de Engenharia Mecânica).
- Buttons:** A red "STOP" button, a grey "LOGIN DO UTILIZADOR" button, and a red "PARAR" button.
- Signal Output Section:** "Saída de sinal em:" with radio buttons for "Deslocamento" (selected), "Velocidade", and "Aceleração". It includes a dropdown for "Unidades Deslocamento" set to "mm" and an "Alarmes?" indicator.
- Graphs:**
 - "Sinal no tempo": A plot of Amplitude vs. Tempo (0 to 0.5) showing a signal fluctuating around zero.
 - "Factor de Crista": A plot of Amplitude vs. Factor de Crista (0 to 14) showing a flat line at zero.
- Cursor Data:** A table showing cursor coordinates: X: 0.51178, Y: -0.0338.

Anexo B

Painel Frontal do sub-VI de palavra-passe incorrecta (tente outra vez):

Palavra-passe **Incorrecta**
Por favor, tente outra vez.

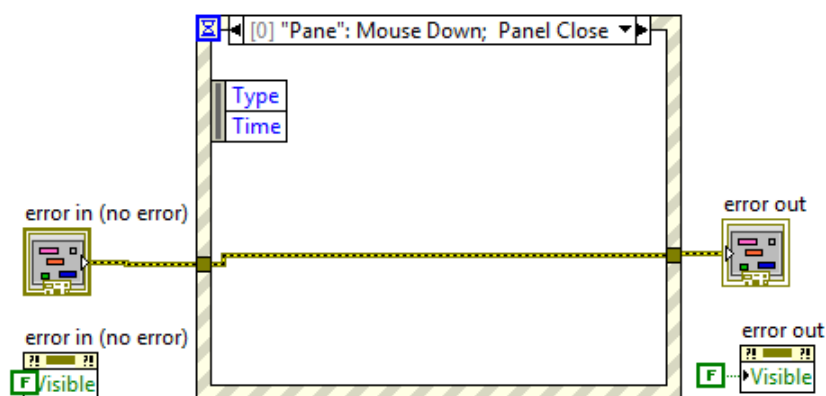


Painel Frontal do sub-VI de palavra-passe incorrecta (número máximo de tentativas atingido):

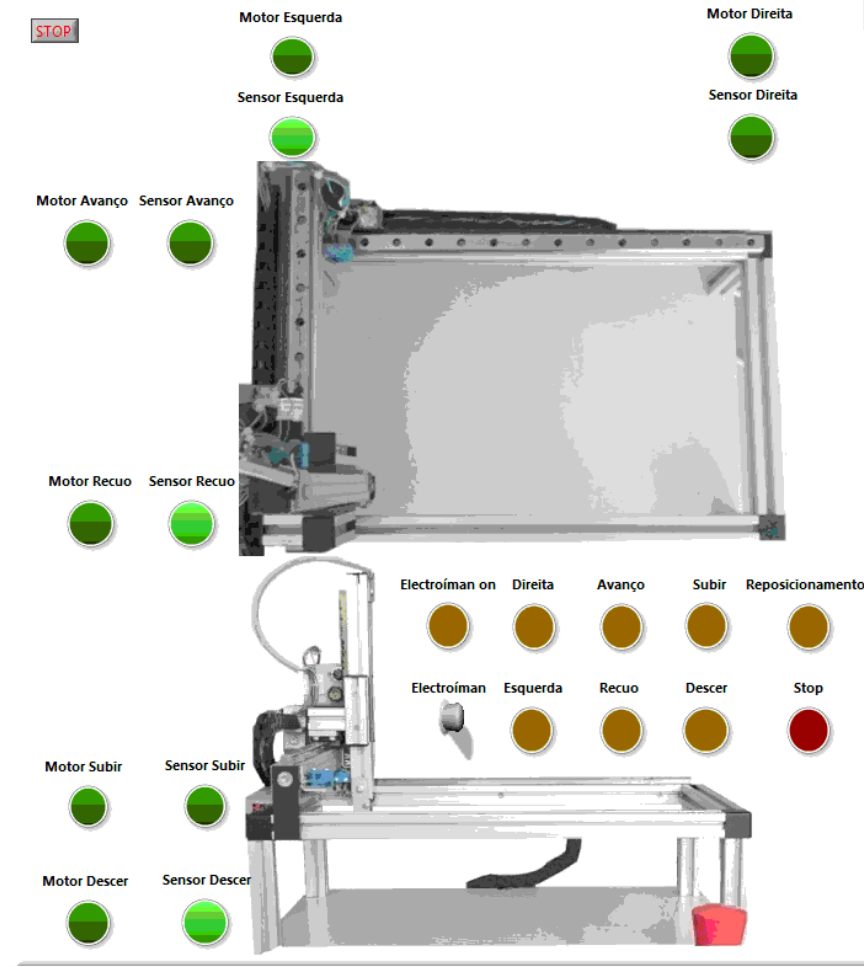
Palavra-passe **Incorrecta**
Alcançou o número máximo de tentativas de login.
Esta aplicação vai encerrar.



Diagrama de Blocos do sub-VI de palavra-passe incorrecta (tente outra vez e número máximo de tentativas atingido):



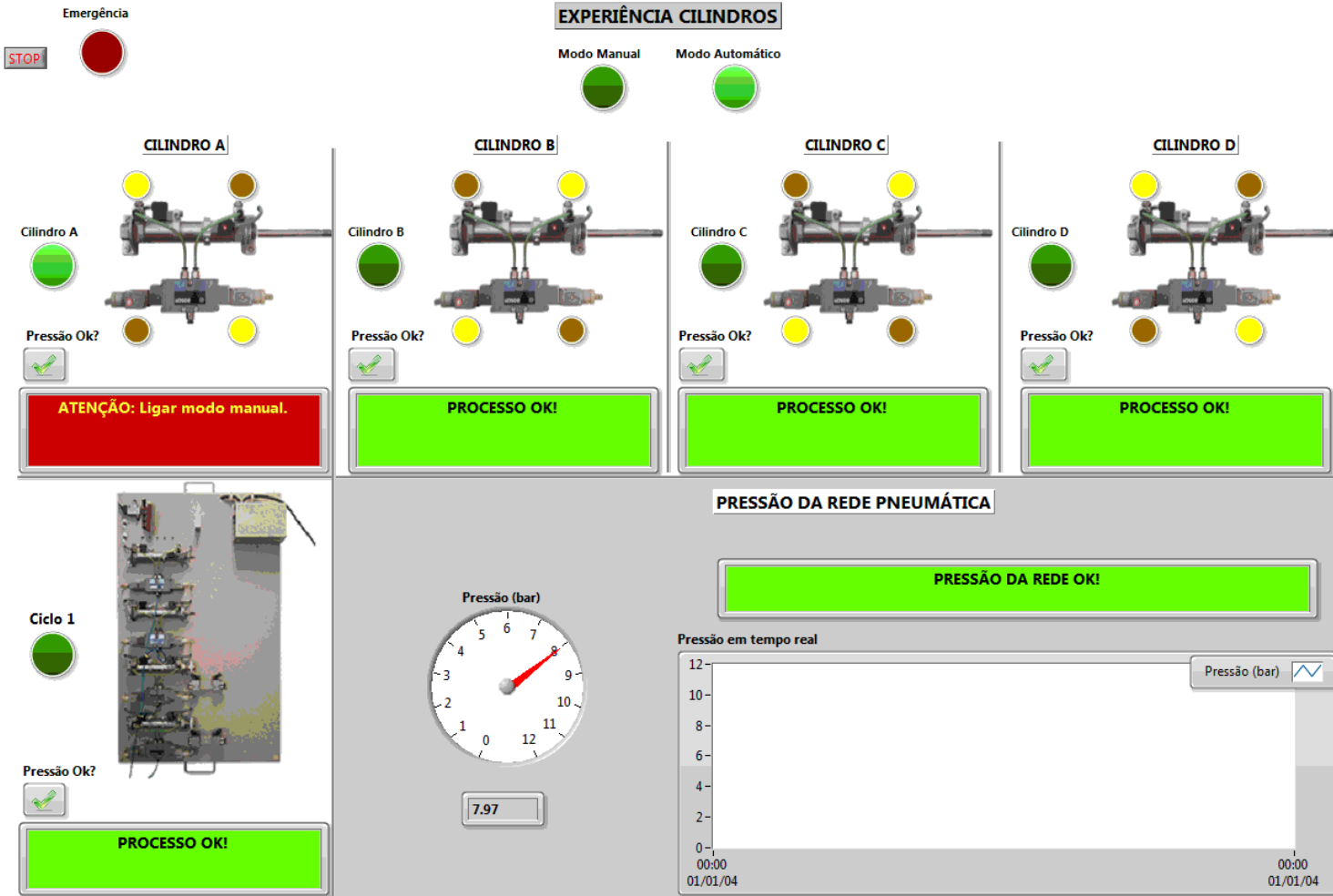
Painel Frontal da Experiência “Pórtico”:





EXPERIÊNCIA PÓRTICO

Electroíman	Pórtico na posição "descida"?
ATENÇÃO: Avaria no sensor "descer". Por favor, encerre a sessão e contacte o administrador.	Sim <input type="button" value="v"/>
Avanço	Pórtico na posição pretendida?
PROCESSO OK!	Não <input type="button" value="v"/>
Recuo	Pórtico na posição pretendida?
PROCESSO OK!	Não <input type="button" value="v"/>
Direita	Pórtico na posição pretendida?
PROCESSO OK!	Não <input type="button" value="v"/>
Esquerda	Pórtico na posição pretendida?
PROCESSO OK!	Não <input type="button" value="v"/>
Subir	Pórtico na posição pretendida?
PROCESSO OK!	Não <input type="button" value="v"/>
Descer	Pórtico na posição pretendida?
PROCESSO OK!	Não <input type="button" value="v"/>

Painel Frontal da Experiência “Cilindros”:



Painel Frontal do Sistema de Análise de Vibrações:

STOP

LOGIN DO UTILIZADOR

PARAR

TIPO DE AQUISIÇÃO DE SINAL

SINAL SIMULADO SINAL ADQUIRIDO

Unidades Deslocamento

Deslocamento: mm/mm

Velocidade: mm/s

Aceleração: mm/s²

Sensibilidade

Ganho:

Vizualização dos gráficos:

Todos

Nº de Harmónicas

Uma

Amplitude	Frequência	Fase
<input type="text" value="50"/>	<input type="text" value="100"/>	<input type="text" value="2"/>
<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>
<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>

Saída de sinal em:

Deslocamento:

Velocidade: mm/s

Aceleração:

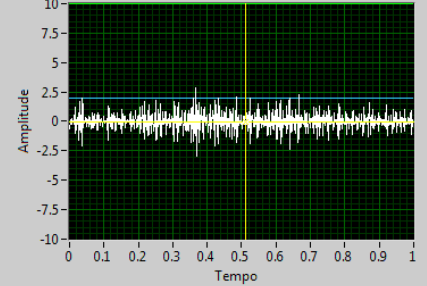
Alarmes?

Alarmes definidos pelo utilizador

Limite Máx: Limite Min:

Limite Máximo: Limite Mínimo:

Sinal no tempo Plot 0



Cursors: X Y
Cursor 0 0.51178 -0.0338

Pico/RMS Logaritmica/Linear

Janelas: Rectangular Pico ou RMS: 2.88437

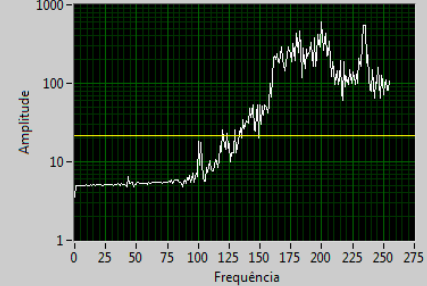
Médias

Nº de Médias:

Ponderação: Exponential

Tipo de Média: No averaging

Espectro FFT Plot 0



Cursors: X Y
Cursor 0 1244.44 21.8841

Factor de Crista

Factor de Crista:

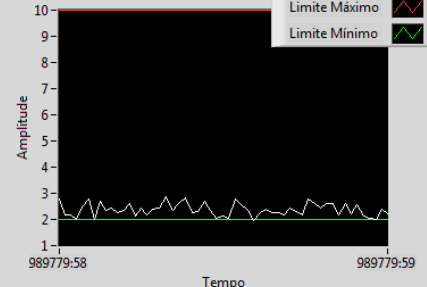
Alarmes?

Alarmes definidos pelo utilizador

Limite Máx: Limite Min:

Limite Máximo: Limite Mínimo:

Factor de Crista Plot 0



Limite Máximo: Limite Mínimo:

Alarmes?

Nível Global

Nível Global: 0.569446

Alarmes do tipo:

Segundo a norma ISO 10816

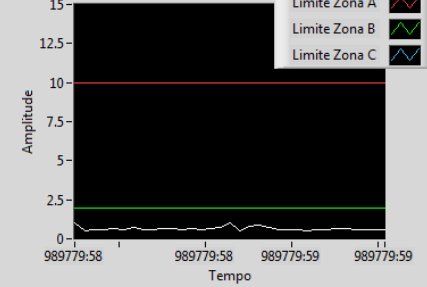
Definidos pelo utilizador

Alarmes definidos pelo utilizador

Limite Máx: Limite Min:

Limite Máximo: Limite Mínimo:

Nível Global Plot 0



Limite Máximo: Limite Mínimo:

Limite Zona A: Limite Zona B: Limite Zona C:

Diagrama de Blocos da Experiência “Pórtico”:

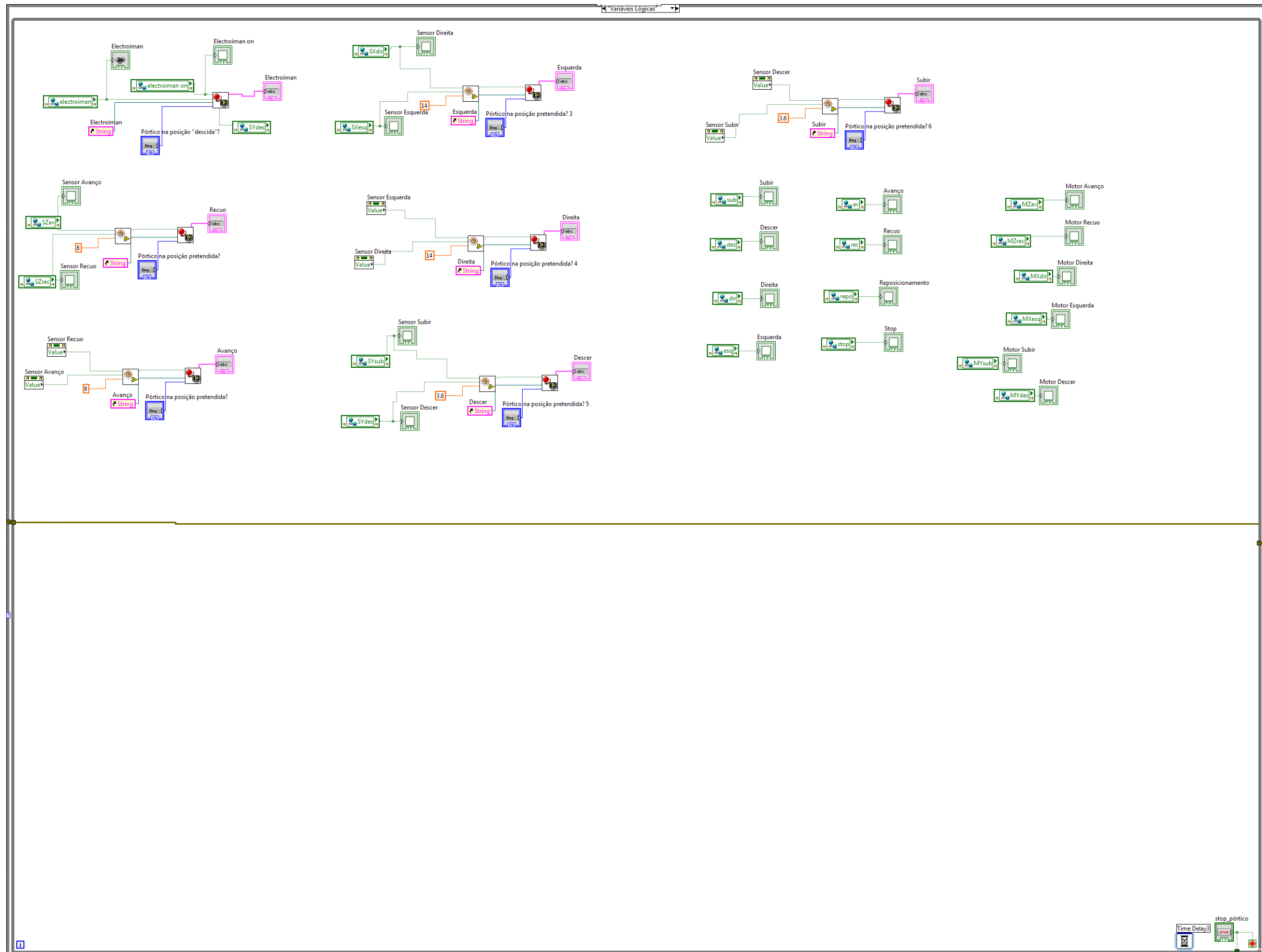


Diagrama de Blocos da Experiência “Cilindros”:

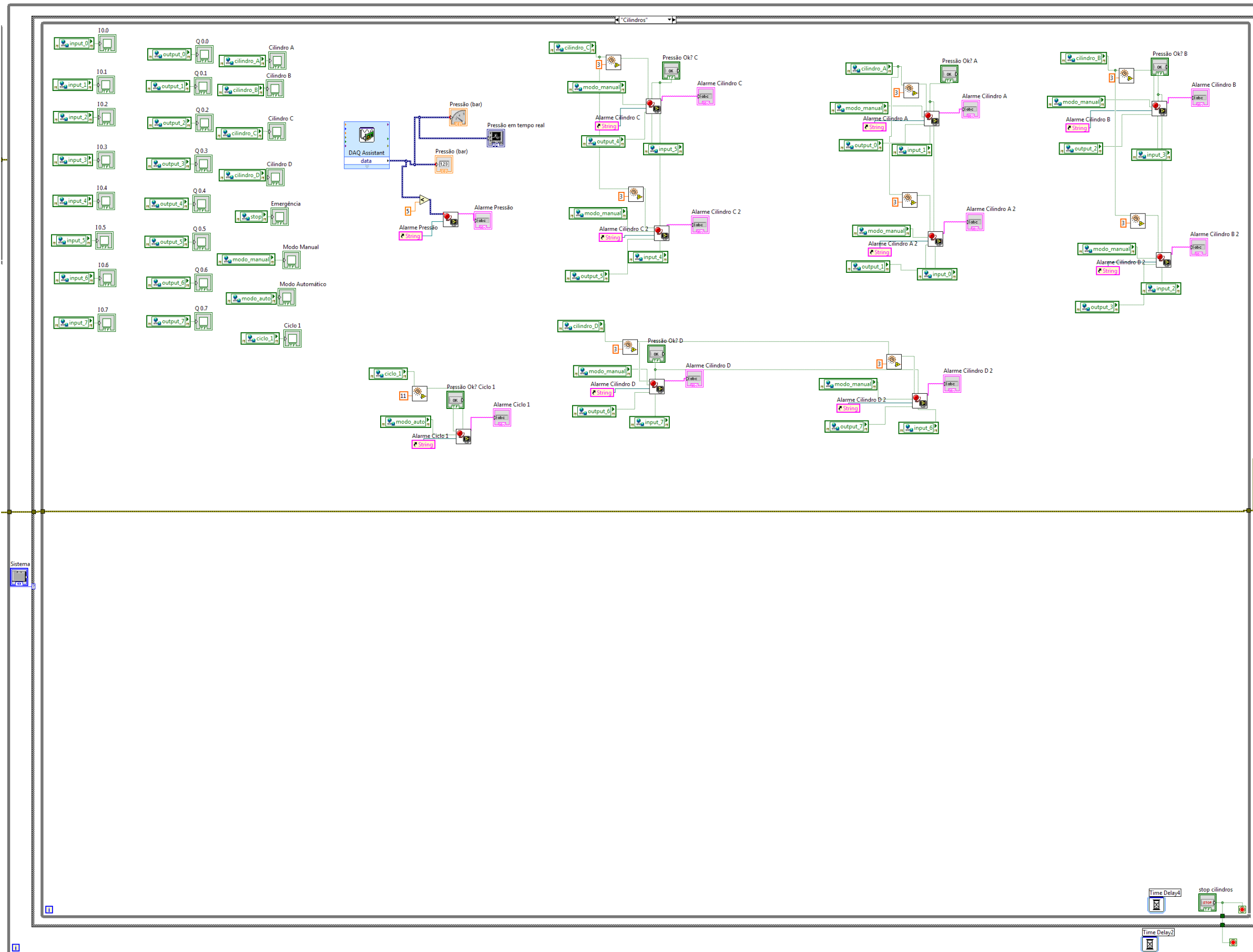


Diagrama de Blocos Sistema de Análise de Vibrações:

