

DETECÇÃO ON-LINE DE DESCONTINUIDADES  
NO PROCESSO DE SOLDAGEM GTAW  
USANDO SENSORIAMENTO INFRAVERMELHO E FPGAs.

RONALD HERNAN HURTADO VELASCO

DISSERTAÇÃO DE MESTRADO EM SISTEMAS MECATRÔNICOS

UNIVERSIDADE DE BRASÍLIA

FACULDADE DE TECNOLOGIA

# **Livros Grátis**

<http://www.livrosgratis.com.br>

Milhares de livros grátis para download.

**UNIVERSIDADE DE BRASÍLIA**

**FACULDADE DE TECNOLOGIA  
DEPARTAMENTO DE ENGENHARIA MECÂNICA**

**DETECÇÃO ON-LINE DE DESCONTINUIDADES  
NO PROCESSO DE SOLDAGEM GTAW  
USANDO SENSORIAMENTO INFRAVERMELHO E FPGAs.**

**RONALD HERNAN HURTADO VELASCO**

**Orientador: Prof. Dr. Sadek C. Absi Alfaro  
Co-orientador: Prof. Dr. Carlos H. Llanos Quintero**

**DISSERTAÇÃO DE MESTRADO EM SISTEMAS MECATRÔNICOS**

**Publicação: ENM.DM-37/10**

**Brasília, 25 de Outubro de 2010**

UNIVERSIDADE DE BRASÍLIA

Faculdade de Tecnologia

DISSERTAÇÃO DE MESTRADO EM SISTEMAS MECATRÔNICOS

**DETECÇÃO ON-LINE DE DESCONTINUIDADES  
NO PROCESSO DE SOLDAGEM GTAW  
USANDO SENSORIAMENTO INFRAVERMELHO E FPGAs.**

**RONALD HERNAN HURTADO VELASCO**

*Dissertação de Mestrado submetida ao Departamento de Engenharia*

*Mecânica da faculdade de Tecnologia da Universidade de Brasília*

*como requisito parcial para a obtenção do grau de Mestre em Sistemas Mecatrônicos*

**Banca Examinadora**

**Prof. Dr. Sadek C. Absi Alfaro,** \_\_\_\_\_

**ENM/UnB**  
*Orientador*

**Prof. Dr. Carlos H. Llanos Quintero,** \_\_\_\_\_

**ENM/UnB**  
*Co-orientador*

**Prof. Dr. Gilmar Ferreira Batalha,** \_\_\_\_\_

**ENM/EPUSP**  
*Examinador externo*

**Prof. Dr. Ricardo Pezzuol Jacobi,** \_\_\_\_\_

**CIC/UnB**  
*Examinador interno*

**Prof. Dr. Guilherme Caribé Carva-** \_\_\_\_\_

**lho, ENM/UnB**  
*Suplente*

Brasília, 25 de Outubro de 2010

## FICHA CATALOGRÁFICA

HURTADO V., Ronald Hernan

Detecção *on-line* de descontinuidades no processo de soldagem GTAW usando sensoriamento infravermelho e FPGAs. [Distrito Federal] 2010.

xv, 85p. 210 × 297 mm (ENM/FT/UnB, Mestre, Sistemas Mecatrônicos, 2010). Dissertação de Mestrado – Universidade de Brasília. Faculdade de Tecnologia.

Departamento de Engenharia Mecânica.

- |                            |                             |
|----------------------------|-----------------------------|
| 1. Detecção <i>on-line</i> | 2. FPGAs                    |
| 3. GTAW                    | 4. Processadores embarcados |
| I. ENM/FT/UnB              | II. Título (série)          |

## REFERÊNCIA BIBLIOGRÁFICA

HURTADO, RONALD. (2010). Detecção *on-line* de descontinuidades no processo de soldagem GTAW usando sensoriamento infravermelho e FPGAs. Dissertação de Mestrado em Sistemas Mecatrônicos, Publicação ENM.DM-37/10, Departamento de Engenharia Mecânica, Universidade de Brasília, Brasília, DF, 85p.

## CESSÃO DE DIREITOS

**AUTOR:** Ronald Hernan Hurtado Velasco.

**TÍTULO:** Detecção *on-line* de descontinuidades no processo de soldagem GTAW usando sensoriamento infravermelho e FPGAs.

**GRAU:** Mestre                      **ANO:** 2010

É concedida à Universidade de Brasília permissão para reproduzir cópias desta dissertação de mestrado e para emprestar ou vender tais cópias somente para propósitos acadêmicos e científicos. O autor reserva outros direitos de publicação e nenhuma parte dessa dissertação de mestrado pode ser reproduzida sem autorização por escrito do autor.

*Aos meus pais, Jose e Omaira e aos meus irmãos, Paulo, Andres e Jose.  
À Andrea.*

RONALD HERNAN HURTADO VELASCO

## Agradecimentos

*Quero agradecer aos meus pais, Jose e Omaira, e aos meus irmãos Paulo, Andres e Jose pelo apoio e motivação.*

*À Andrea pelo apoio, amizade e amor incondicional.*

*À professora Sonia pela confiança e amizade.*

*Ao meu orientador Prof. Sadek A. Alfaro, pela confiança e a possibilidade de trabalhar no seu lado.*

*Ao Prof. Carlos Llanos Quintero, pela confiança e pela ajuda no desenvolvimento desse trabalho*

*Aos colegas e amigos, Claudia, Ana M., Diego M., Alvaro, Rodrigo, Fernand, Yesid, Diego F., Liliana, Jesus, Janier, Jones, Edgar, Daniel, Andre, Thiago, ao técnico Antônio Marrocos e à Ana R. E a todos meus amigos e colegas na universidade como fora dela.*

*Ao Grupo de Automação e Controle (GRACO).*

*Por fim agradeço à Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) pelo apoio financeiro.*

RONALD HERNAN HURTADO VELASCO

# RESUMO

## DETECÇÃO ON-LINE DE DESCONTINUIDADES NO PROCESSO DE SOLDAGEM GTAW USANDO SENSORIAMENTO INFRAVERMELHO E FPGAs.

**RONALD HERNAN HURTADO VELASCO.**

Programa de pós-graduação em sistemas mecatrônicos.

Universidade de Brasília.

Este trabalho apresenta o desenvolvimento e a implementação de um sistema embarcado, baseado em FPGA para o monitoramento *on-line* do arco voltaico no processo de soldagem GTAW, utilizando um algoritmo de detecção de mudança. Este monitoramento permitiu detectar a presença de descontinuidades durante o processo de soldagem, utilizando um sensor de radiação infravermelha.

O algoritmo de detecção de mudança, descrito em hardware usando VHDL, possui um filtro Kalman e uma regra de parada com um teste estatístico por somas cumulativas (CUSUM). Todos os cálculos para a detecção de descontinuidades foram feitos em aritmética de ponto flutuante de 32 bit. As simulações das descrições hardware foram feitas no software ModelSIM. O dispositivo reconfigurável utilizado para embarcar o algoritmo de detecção de mudança e outros periféricos do sistema foi o FPGA Spartan-3E 1600E da Xilinx. Para o gerenciamento do algoritmo e os outros periféricos embarcados, utilizou-se o processador *soft MicroBlaze*.

Inicialmente, para detectar descontinuidades foi necessário obter o padrão da solda desejada. Para isto foram realizados alguns teste de soldagem sem a presença aparente de descontinuidades, variando-se a distância do eletrodo à peça, bem como a corrente de soldagem. Posteriormente, foram realizados testes com a presença de descontinuidades. Os tipos de descontinuidades avaliadas no trabalho foram: presença de arame, furos e areia.

Após a validação através das simulações, o sistema de detecção de descontinuidades foi testado em uma bancada de soldagem com o intuito de avaliar o sucesso na detecção *on-line* de descontinuidades. Durante o processo de soldagem o alarme do sistema foi ativado com cada descontinuidade encontrada. Os dados da detecção de descontinuidades (quantidade e distâncias) para cada teste podem ser armazenados no FPGA ou enviados para o computador. Os resultados permitiram verificar o uso de sistemas embarcados baseados em FPGAs na detecção *on-line* de descontinuidades na soldagem.



# RESUMEN

## DETECCIÓN ON-LINE DE DISCONTINUIDADES EN EL PROCESO DE SOLDADURA GTAW USANDO SENSORIAMENTO INFRAROJO Y FPGAs.

RONALD HERNAN HURTADO VELASCO.

Programa de pós-graduação em sistemas mecatrônicos.

Universidade de Brasília.

Este trabajo presenta el diseño e implementación de un sistema embarcado basado en FPGA para el monitoreo *on-line* del arco voltaico en el proceso de soldadura GTAW, usando un algoritmo de detección de cambios. Este monitoreo permitió detectar la presencia de discontinuidades durante el proceso de soldadura utilizando un sensor de radiación infrarroja.

El algoritmo de detección de cambios, descrito en hardware usando VHDL, posee un filtro Kalman y una regla de parada con un test estadístico por sumas acumulativas (CUSUM). Todos los cálculos para la detección de discontinuidades fueron hechos en aritmética de punto flotante de 32 bit. Las simulaciones de las descripciones hardware fueron hechas en ModelSM. El dispositivo reconfigurable utilizado para embarcar el algoritmo de detección de cambios y los demás periféricos del sistema fue el FPGA Spartan 3E 1600 de Xilinx. Para el gerenciamiento del algoritmo y los otros periféricos embarcado, se utilizó el procesador *soft MicroBlaze*.

Inicialmente, para detectar discontinuidades fue necesario obtener el patrón de soldadura deseada. Para esto se realizaron algunos test de soldadura sin presencia aparente de discontinuidades, variando la distancia del electrodo a la pieza y la corriente de soldadura. Posteriormente, fueron realizados test con la presencia de discontinuidades. Los tipos de discontinuidades validadas en este trabajo fueron: presencia de alambre, huecos y arena.

Después de la validación a través de las simulaciones, el sistema de detección de discontinuidades fue probado en el montaje de soldadura con el propósito de validar el éxito en la detección *on-line* de discontinuidades. Durante el proceso de soldadura la alarma del sistema fue activada con cada discontinuidad encontrada. Los datos de la detección de discontinuidades (cantidad y posición) para cada test pueden ser almacenados en el FPGA o enviados a un computador. Los resultados permitieron verificar el uso de sistemas embarcados basados en FPGAs en la detección *on-line* de discontinuidades en soldadura.

# ABSTRACT

## ON-LINE DETECTION OF DISCONTINUITIES IN THE GTAW WELDING PROCESS USING INFRARED SENSING AND FPGAs.

RONALD HERNAN HURTADO VELASCO.

Programa de pós-graduação em sistemas mecatrônicos.

Universidade de Brasília.

This work presents the development and implementation of an FPGA-based embedded system to *on-line* arc welding monitoring in a GTAW process, using a change detection algorithm. The monitoring allowed the discontinuities detection during welding process using an infrared sensor.

The change detection algorithm, described in hardware using VHDL, has a Kalman filter and a stopping rule with a statistical test by cumulative sums (CUSUM). All the calculations for discontinuities detection were done in 32 bit floating point arithmetic. The simulations of hardware descriptions were made using ModelSim software. Spartan-3E 1600E FPGA from Xilinx was used to embedded the algorithm and the other peripherals of the system. To perform the management of the algorithm and the others embedded devices, the processor *soft MicroBlaze* was used.

At first, it is necessary to obtain a desired welding pattern in order to be able to detect discontinuities. Several tests without apparent presence of discontinuities were performed, changing the distance from the electrode to the work piece and the welding current. Then, tests were performed in presence of discontinuities. The different types of discontinuities used during welding were presence of wire, holes and sand.

Once the system was validated through simulations, the discontinuities detection system was tested in the welding bench in order to evaluate the on-line discontinuities detection effectiveness. As a result, during the welding process the system alarm was activated with each discontinuity that was found. Discontinuity detection data (quantity and location) for each test can be stored in the FPGA or sent to a PC. Obtained results had verified the use of FPGA-based embedded system to *on-line* discontinuities detection during welding processes.

# SUMÁRIO

|   |            |
|---|------------|
| <b>Lista de Figuras</b> .....                   | <b>xiv</b> |
| <b>Lista de Tabelas</b> .....                   | <b>xv</b>  |
| <b>1 Introdução</b> .....                       | <b>1</b>   |
| 1.1 Objetivos do trabalho. ....                 | 2          |
| 1.1.1 Objetivo geral. ....                      | 2          |
| 1.1.2 Objetivos específicos. ....               | 2          |
| 1.2 Justificativa do trabalho. ....             | 3          |
| 1.3 Apresentação do trabalho.....               | 4          |
| <b>2 Revisão Bibliográfica</b> .....            | <b>5</b>   |
| 2.1 Soldagem.....                               | 5          |
| 2.2 O processo de soldagem GTAW.....            | 6          |
| 2.2.1 Algumas variáveis do processo.....        | 7          |
| 2.2.2 Equipamento.....                          | 8          |
| 2.2.2.1 Fontes.....                             | 8          |
| 2.2.2.2 Tochas. ....                            | 8          |
| 2.2.2.3 Eletrodos.....                          | 8          |
| 2.2.2.4 Gás de proteção.....                    | 9          |
| 2.3 Sensoriamento infravermelho .....           | 9          |
| 2.3.1 Radiação infravermelha .....              | 9          |
| 2.3.2 Medição da temperatura por radiação ..... | 11         |
| 2.3.2.1 Detectores de radiação térmica .....    | 12         |
| 2.4 O filtro Kalman .....                       | 12         |
| 2.4.1 O algoritmo do filtro Kalman .....        | 14         |
| 2.4.1.1 Atualização do tempo .....              | 14         |
| 2.4.1.2 Atualização da medida .....             | 15         |
| 2.4.2 Parâmetros e ajuste do filtro .....       | 16         |
| 2.5 Detecção de mudança.....                    | 17         |

|          |  |           |
|----------|--|-----------|
| 2.5.1    | Medida da distância .....                    | 17        |
| 2.5.2    | Regra de parada com teste CUSUM.....         | 18        |
| 2.6      | Tecnologia dos FPGAs .....                   | 19        |
| 2.6.1    | Aplicações .....                             | 20        |
| 2.6.2    | Arquitetura dos FPGAs .....                  | 20        |
| 2.6.2.1  | Processadores embarcados.....                | 21        |
| 2.6.2.2  | Processador <i>MicroBlaze</i> .....          | 23        |
| 2.6.2.3  | Blocos de propriedade intelectual (IP) ..... | 24        |
| <b>3</b> | <b>Procedimento e Metodologia .....</b>      | <b>25</b> |
| 3.1      | Sistema de detecção.....                     | 25        |
| 3.1.1    | Co-projeto Hardware/Software.....            | 25        |
| 3.1.2    | Distribuição de tarefas .....                | 26        |
| 3.1.2.1  | Porção sequencial .....                      | 27        |
| 3.1.2.2  | Porção concorrente .....                     | 27        |
| 3.1.3    | Implementações hardware e software .....     | 27        |
| 3.1.3.1  | Implementação hardware .....                 | 27        |
| 3.1.3.2  | Implementação software .....                 | 29        |
| 3.1.4    | Configuração do FPGA .....                   | 29        |
| 3.1.5    | Teste e verificação .....                    | 31        |
| 3.2      | Sistema de soldagem. ....                    | 33        |
| 3.2.1    | Uso do sistema de soldagem. ....             | 36        |
| 3.2.1.1  | Etapa de caracterização.....                 | 36        |
| 3.2.1.2  | Etapa de provas e verificação .....          | 38        |
| 3.3      | Testes para validação. ....                  | 39        |
| <b>4</b> | <b>Resultados .....</b>                      | <b>40</b> |
| 4.1      | Resultados hardware.....                     | 40        |
| 4.1.1    | Conversão análoga/digital.....               | 41        |
| 4.1.2    | Detector de mudanças.....                    | 44        |
| 4.1.2.1  | <b>covariancia&amp;Ganho</b> .....           | 44        |
| 4.1.2.2  | <b>estimador&amp;erro</b> .....              | 45        |
| 4.1.2.3  | <b>CUSUM&amp;limiar</b> .....                | 46        |
| 4.1.3    | Conversão e detecção .....                   | 49        |
| 4.1.4    | Gerenciador do tempo.....                    | 51        |
| 4.1.5    | Memória de descontinuidades.....             | 52        |
| 4.1.6    | Gerenciamento do <i>MicroBlaze</i> .....     | 53        |
| 4.1.6.1  | Entrada/saída de propósito geral.....        | 54        |

|          |  |           |
|----------|--|-----------|
| 4.1.6.2  | RS-232 serial .....  | 54        |
| 4.1.6.3  | RS-232 modem.....  | 54        |
| 4.1.6.4  | LCD.....   | 55        |
| 4.1.6.5  | <i>Timer</i> .....   | 55        |
| 4.1.6.6  | Interrupções.....  | 55        |
| 4.1.7    | Fluxo de execução.....                                     | 56        |
| 4.1.8    | Consumos de recursos.....                                  | 56        |
| 4.2      | Resultados de Soldagem.....                                | 59        |
| 4.2.1    | Curva característica do sistema de soldagem.....           | 60        |
| 4.2.2    | Ajuste do sistema de detecção de mudanças.....             | 61        |
| 4.2.2.1  | Cordão padrão de solda.....                                | 62        |
| 4.2.2.2  | Cordões de soldagem com descontinuidades.....              | 64        |
| 4.2.3    | Validação do sistema de detecção de descontinuidades ..... | 73        |
| 4.3      | Discussão dos resultados.....                              | 78        |
| <b>5</b> | <b>Conclusões e futuros trabalhos .....</b>                | <b>81</b> |
| 5.1      | Conclusões.....  | 81        |
| 5.2      | Futuros trabalhos.....                                     | 82        |
|          | <b>Referências Bibliográficas .....</b>                    | <b>83</b> |

# LISTA DE FIGURAS

|      |  |    |
|------|--|----|
| 2.1  | Classificação de alguns dos processos de soldagem mais importantes. ....   | 5  |
| 2.2  | Esquema essencial do processo GTAW manual. ....  | 7  |
| 2.3  | Tocha GTAW refrigerada por água. ....  | 9  |
| 2.4  | Esquema do espectro eletromagnético. ....  | 10 |
| 2.5  | Distribuição espectral de Plank para um corpo negro. Têm plotadas as temperaturas para a temperatura ambiente (300 K) e a temperatura de ebulição da água (373.15 K). ....   | 11 |
| 2.6  | Diagrama de blocos de um pirômetro automático. ....  | 12 |
| 2.7  | Esquema geral do algoritmo do filtro Kalman ....   | 16 |
| 2.8  | Passos do modelo geral para um processo de detecção de mudança. ....   | 17 |
| 2.9  | Estrutura mais comum da arquitetura de um FPGA. ....   | 21 |
| 2.10 | Diagrama de blocos do core do <i>MicroBlaze</i> . ....   | 24 |
| 3.1  | Diagrama de blocos do procedimento usado para desenvolver o sistema de detecção de descontinuidades com o FPGA. ....   | 26 |
| 3.2  | Ligação entre o processador <i>MicroBlaze</i> e periféricos personalizados criados em VHDL através do barramento PLB. ....   | 31 |
| 3.3  | Esquema do montagem para verificação e teste do sistema de detecção. ....  | 32 |
| 3.4  | Diagrama esquemático do sistema de soldagem. ....  | 33 |
| 3.5  | Fotografias que apresentam o sistema de soldagem e como a suas partes estão dispostas. ....  | 35 |
| 3.6  | Bancada para obter a curva característica do arco GTAW. ....   | 37 |
| 4.1  | Esquema que representa os periféricos de soma/subtração multiplicação e divisão usados em alguns dos periféricos desenvolvidos. Relembrando que o periférico de soma e subtração é o mesmo, o tipo de operação é definido por um sinal de entrada, que para simplificar não aparece desenhada nos esquemas. .... | 41 |
| 4.2  | Diagrama da arquitetura do periférico <b>conversaoAD</b> , que faz a conversão analógica/digital do sinal do sensor infravermelho e entrega o resultado em 32 bit. ....  | 42 |

|      |   |    |
|------|---|----|
| 4.3  | Diagrama de resposta no tempo para o periférico <b>conversaoAD</b> , o tempo consumido desde a ativação do periférico até que o resultado esta disponível é $t_{10} = 3311$ ns. ....  | 42 |
| 4.4  | Diagrama da arquitetura do periférico <b>covariancia&amp;Ganho</b> que é ativado ou por o sinal de entrada <b>En_grl</b> ou por o sinal de entrada <b>proxGanho</b> , e gera o valor da contante do filtro no sinal de saída <b>KGanho</b> . ....                   | 44 |
| 4.5  | Diagrama da arquitetura do periférico <b>estimador&amp;erro</b> que é ativado por <b>dadoSet</b> e <b>GanhoSet</b> e gera o valor estimado do filtro estimado. ....   | 46 |
| 4.6  | Diagrama da arquitetura do periférico <b>CUSUM&amp;limiar</b> que é ativado por o sinal de entrada <b>erroSet</b> para gerar a alarme, se houver, a partir do sinal de entrada <b>erro</b> . ....   | 47 |
| 4.7  | Diagrama de blocos do periférico <b>Detector</b> que apresenta a maneira como os periféricos <b>covariancia&amp;Ganho</b> , <b>estimador&amp;erro</b> e <b>CUSUM&amp;limiar</b> foram interligados. ....  | 48 |
| 4.8  | Diagrama de blocos do periférico <b>Detector</b> visto como uma estrutura <i>pipeline</i> de dois estágios. ....  | 48 |
| 4.9  | Diagrama de resposta no tempo para o periférico <b>Detector</b> , o tempo consumido desde a ativação do periférico pelo flanco de decida do sinal <b>dadoSet</b> até que o resultado esta disponível é $t_{10} = 850$ ns. ....                                      | 49 |
| 4.10 | Diagrama da arquitetura do periférico <b>conversaoDetetor</b> . Contem como sub-periféricos <b>conversaoAD</b> , <b>Detector</b> e <b>monoestavel</b> ....  | 49 |
| 4.11 | Diagrama de resposta no tempo para do sinal <b>limiarSet</b> que indica o tempo consumido desde o início da conversão analógica/digital até a estimativa. ....  | 51 |
| 4.12 | Diagrama da arquitetura do periférico <b>gerTempo</b> que é ativado pelo sinal de entrada <b>En</b> e conta o tempo do processo de soldagem ....  | 52 |
| 4.13 | Diagrama da arquitetura do periférico <b>memRAM</b> que armazena a quantidade de descontinuidades e onde elas aconteceram. ....   | 53 |
| 4.14 | Diagrama da arquitetura do sistema de detecção de descontinuidades que apresenta o <i>MicroBlaze</i> e os periféricos envolvidos. ....  | 56 |
| 4.15 | Fluxo de execução na detecção de descontinuidades. Estão incluído o software e o hardware. As execuções no hardware são concorrentes. ....  | 57 |
| 4.16 | Curva característica do arco GTAW. A faixa de corrente foi 30-200 A e o <i>stand-off</i> foi 3 e 5 mm. ....   | 60 |
| 4.17 | Curva de resposta do sensor de infravermelho para a caracterização do arco GTAW, com uma faixa de corrente de 30-200 A e <i>stand-off</i> de 3 e 5 mm. ....   | 61 |
| 4.18 | Cordão de solda padrão sem descontinuidades aparentes que foi conseguido a 140 A, e com um <i>stand-off</i> de 5 mm. De acima para abaixo na figura estão a tensão do arco, a corrente do arco, a tensão do sensor infravermelho e o aspecto que o cordão tem. .... | 62 |

|      |   |    |
|------|---|----|
| 4.19 | Teste de normalidade dos resíduos do sinal do sensor infravermelho. (a) 140 A, <i>stand-off</i> = 3 mm, $p = 0,99999$ (b) 140 A, <i>stand-off</i> = 5 mm, $p = 0,99999$ (c) 140 A, <i>stand-off</i> = 3 mm, $p = 0,99998$ (d) 140 A, <i>stand-off</i> = 5 mm, $p = 0,99999$ | 65 |
| 4.20 | Cordão de solda com descontinuidades produzidas por furos, conseguido a 140 A e com um <i>stand-off</i> de 5 mm. De acima para abaixo na figura estão a tensão do arco, a corrente do arco, a tensão do sensor infravermelho e o aspecto que o cordão tem. ....             | 66 |
| 4.21 | Resultado do ajuste para a detecção de descontinuidades por arame. <i>stand-off</i> = 3 mm e descontinuidades no par distância/quantidade-de-aramé: 35 mm/1, 45 mm/2, 75 mm/3 e 105 mm/1.....   | 67 |
| 4.22 | Resultado do ajuste para a detecção de descontinuidades por arame. <i>stand-off</i> = 5 mm e descontinuidades no par distância/quantidade-de-aramé: 45 mm/3, 75 mm/3 e 105 mm/1. ....   | 68 |
| 4.23 | Resultado do ajuste para a detecção de descontinuidades por furos. <i>stand-off</i> = 3 mm e descontinuidades em: 19-28 mm, 52-59 mm e 80-90 mm. ....   | 69 |
| 4.24 | Resultado do ajuste para a detecção de descontinuidades por furos. <i>stand-off</i> = 5 mm e descontinuidades em: 21-27 mm, 50-58 mm e 78-86 mm. ....   | 70 |
| 4.25 | Resultado do ajuste para a detecção de descontinuidades por presença de areia. <i>stand-off</i> = 3 mm e descontinuidades em: 21-30 mm, 60-70 mm e 90-105 mm.....   | 71 |
| 4.26 | Resultado do ajuste para a detecção de descontinuidades por presença de areia. <i>stand-off</i> = 5 mm e descontinuidades em: 22-35 mm, 55-70 mm e 90-105 mm.....   | 72 |
| 4.27 | Resultado da detecção <i>on-line</i> de descontinuidades geradas por arame. <i>stand-off</i> = 3 mm e descontinuidades no par distância/quantidade-de-aramé: 35 mm/1, 45 mm/2, 75 mm/3 e 105mm/1.....   | 74 |
| 4.28 | Resultado da detecção <i>on-line</i> de descontinuidades geradas por arame. <i>stand-off</i> = 5 mm e descontinuidades no par distância/quantidade-de-aramé: 25 mm/1, 45 mm/2, 80 mm/3 e 100 mm/1.....  | 75 |
| 4.29 | Resultado da detecção <i>on-line</i> de descontinuidades geradas por furos. <i>stand-off</i> = 3 mm e descontinuidades entre: 19-28 mm, 52-59 mm e 80-90 mm. ....   | 76 |
| 4.30 | Resultado da detecção <i>on-line</i> de descontinuidades geradas por furos. <i>stand-off</i> = 5 mm e descontinuidades em: 21-27 mm, 50-58 mm e 78-86 mm. ....  | 77 |
| 4.31 | Resultado da detecção <i>on-line</i> de descontinuidades geradas por presença de areia. <i>stand-off</i> = 3 mm e descontinuidades entre: 21-30 mm, 60-70 mm e 90-105 mm....  | 78 |
| 4.32 | Resultado da detecção <i>on-line</i> de descontinuidades geradas por presença de areia. <i>stand-off</i> = 5 mm e descontinuidades entre: 22-35 mm, 55-70 mm e 90-105 mm....  | 79 |



## LISTA DE TABELAS

|     |   |    |
|-----|---|----|
| 3.1 | Alguns recursos da FPGA Spartan-3E 1600 (XC3S1600E). .....  | 30 |
| 3.2 | Algumas especificações técnicas do sensor TL-GA-18-1 .....  | 36 |
| 3.3 | Parâmetros de soldagem para os testes de validação. ....  | 39 |
| 4.1 | Descrição dos tempos consumidos pelos sinais mais relevantes do periférico <b>con-</b><br><b>versaoAD</b> . .....   | 43 |
| 4.2 | Descrição dos tempos consumidos pelos sinais mais relevantes do periférico <b>Detector</b> . ..   | 50 |
| 4.3 | Relação de consumo de recursos para o <i>MicroBlaze</i> e os periféricos de propósito<br>geral usados. ....   | 58 |
| 4.4 | Relação de consumo de recursos para os periféricos personalizados. ....   | 58 |
| 4.5 | Relação de consumo recursos para os dispositivos extras usados pelo <i>MicroBlaze</i> . ..  | 58 |
| 4.6 | Consumo total de recursos do FPGA.....  | 59 |
| 4.7 | Máximas frequências de operação calculadas. ....  | 59 |
| 4.8 | Resposta do sensor infravermelho no processo de obter o padrão de solda, os parâ-<br>metros corrente e <i>stand-off</i> são mudados, em quanto a vazão de gás e velocidade<br>permanecem constantes. .... | 63 |
| 4.9 | Valores da variância para cada cordão sem descontinuidades usados para obter o<br>padrão. O valor médio para estes dados é $R = 0.0029$ .....   | 64 |

# 1 INTRODUÇÃO

Atualmente nos laboratórios do Grupo de Automação e Controle (GRACO) da Universidade de Brasília é realizada uma grande variedade de projetos de pesquisa enfocados na soldagem e dispositivos reconfiguráveis, entre outras áreas. A pesquisa em soldagem está empenhada no estudo do processo de soldagem como a monitoração da qualidade da solda. Os principais processos utilizados são: a Soldagem a Arco-Gás com eletrodo de Tungstênio (*Gas Tungsten Arc Welding* – GTAW), que também é o processo de soldadura utilizado para o presente trabalho, e o processo de Soldagem a Arco-Gás Metal (*Gas Metal Arc Welding* – GMAW). Os estudos da soldagem são feitos mediante o uso de uma variedade de sensores, que incluem: *pirômetros*, *câmeras de alta velocidade* e de *infravermelho*, *sonômetros* e *espectrômetros*. Também são medidas a corrente e tensão durante a soldagem. As análises destes dados é realizada, sobretudo de maneira *off-line*, o que significa que os dados só serão analisados depois da realização do processo de soldagem. A análise *off-line* realizada é principalmente estatística e utiliza ferramentas de software como o LabVIEW e o MatLab. A pesquisa realizada no GRACO na área de dispositivos reconfiguráveis baseados em FPGA (*Field Programmable Gate Array*) encontra-se dirigida ao estudo de uma variedade de algoritmos e métodos para sua implementação, otimização hardware e sua simulação, nos quais se aproveitam as capacidades de paralelismo intrínsecas tanto dos algoritmos (se for o caso) como do FPGA. As áreas em que as FPGAs foram utilizadas incluem as seguintes: redes neurais, lógica *fuzzy*, automação e controle, robótica, automação predial e processamento de imagens. Também se tem trabalhado em projetos com enfoque hardware/software, que envolvem os processadores *Soft* como o *MicroBlaze* da Xilinx e o Nios da Altera.

Esta pesquisa apresenta uma aplicação dos dispositivos reconfiguráveis —FPGAs— na área de soldagem, mais precisamente a área de monitoração da qualidade da solda. A aplicação consistiu no desenvolvimento do sistema de monitoramento *on-line* do processo de Soldagem a Arco-Gás com eletrodo de Tungstênio (GTAW), através do sensoriamento óptico da radiação infravermelha de seu arco voltaico como meio para detectar discontinuidades que a soldagem poderia apresentar. O sistema de monitoramento foi abordado usando uma metodologia de implementação para um sistema embarcado em FPGA, em que foram utilizadas técnicas de co-projeto Hardware/Software (*Hardware/Software co-design*). Na parte hardware encontram-se descritos os periféricos que

contêm o algoritmo para a detecção de mudanças, composto principalmente, pelo módulo para o filtro Kalman e pelo módulo do teste estatístico por somas cumulativas (CUSUM). A parte software está contida no processador *Soft MicroBlaze* e tem a função de gerenciamento de todos os periféricos e interface de usuário.

Por fim, mostra-se que o sistema desenvolvido em hardware é adequado para o propósito da detecção *on-line* de descontinuidades no processo de soldagem.

## 1.1 OBJETIVOS DO TRABALHO.

### 1.1.1 Objetivo geral.

Desenvolver um sistema para a detecção *on-line* das descontinuidades no processo de Soldagem a Arco-Gás com eletrodo de Tungstênio (GTAW), mediante o sensoriamento óptico da radiação infravermelha do seu arco voltaico usando um algoritmo de detecção de mudança descrito em hardware e embarcado em um FPGA.

### 1.1.2 Objetivos específicos.

- Aproveitar e utilizar as características intrínsecas de paralelismo tanto do algoritmo de detecção de mudança como do processo de soldagem no desenvolvimento do sistema de detecção de descontinuidades.
- Desenvolver um sistema baseado na metodologia de co-projeto Hardware/Software (*Hardware/Software co-design*) para o controle da bancada de soldagem GTAW (sistema de aquisição de dados, mesa posicionadora e fonte de soldagem).
- Continuar a pesquisa na monitoração do processo de soldagem GTAW mas usando desta vez como nova abordagem o FPGA.
- Encaminhar as pesquisas realizadas no laboratório de soldagem do GRACO no campo da monitoração *on-line*.
- Incluir processadores embarcados em FPGA do tipo *Soft* para gerenciar os periféricos descritos em hardware do algoritmo de detecção e a interface de usuário do sistema para a detecção das descontinuidades.

## 1.2 JUSTIFICATIVA DO TRABALHO.

São muitas as variáveis presentes que envolvem os processos de soldagem, incluindo o processo de soldagem por arco elétrico. Uma destas variáveis são as altas temperaturas atingidas, que por sua vez é característica intrínseca dos processos de soldagem. Esta característica do processo pressupõe o uso de sensoriamento da temperatura como uma benéfica opção para identificar as descontinuidades na solda quando apresentar variações da temperatura ou variações na radiação infravermelha, sendo o uso de sensores não invasivos ou de não contato, como os sensores ópticos de infravermelho utilizados nesta pesquisa, o sensoriamento adequado para este tipo de ambiente. O sensoriamento infravermelho já tem sido provado com sucesso no processo de soldagem GTAW na detecção de descontinuidades [1].

Tanto na soldagem industrial como ao nível acadêmico, é relevante detectar descontinuidades para descartar os possíveis falhas que possam colocar em risco a integridade de uma estrutura ou de uma peça. Em termos gerais existem duas técnicas de monitoração em soldagem usadas para detectar as descontinuidades: técnicas *off-line* e *on-line*. Na técnica *off-line*, o processo de monitoração é realizado depois que a solda foi composta, sendo esta, a técnica de monitoração mais utilizada, tanto na indústria como na pesquisa acadêmica. Por outro lado, a técnica de monitoração *on-line* permite detectar as descontinuidades enquanto elas estão ocorrendo, permitindo que o sistema de soldagem faça as correções necessárias, e minimizando as falhas que as descontinuidades provocam, além de reduzir os custos do processo.

Com o propósito de aperfeiçoar o sistema de monitoramento *off-line* para o *on-line*, muitas alternativas têm sido propostas, como as propostas em [2, 3]. Entretanto, todas estas técnicas fazem uso de uma plataforma PC, na que os cálculos são feitos de maneira sequencial, além de ter a dependência do sistema operativo que introduz uma baixa no rendimento do processador.

Dado o fato de que a técnica do monitoração *on-line* precisa de um sistema que responda de forma apropriada, garantindo uma resposta suficientemente rápida, implica que o algoritmo usado para detectar descontinuidades precisa ter um bom desempenho e uma boa plataforma. Desta maneira, o sistema de monitoramento *on-line* possibilita ser implementado em um sistema embarcado, conseguindo um bom desempenho e baixo consumo de potência, além de dar a opção de portabilidade.

Com esta pesquisa propõe-se o uso de um sistema embarcado baseado em FPGA como uma ferramenta alternativa para aproximar-se do monitoramento *on-line*. Os sistemas embarcados projetados em FPGAs proporcionam flexibilidade, alto desempenho e reprogramabilidade para a implementação de aplicações eletrônicas. Além disso, os FPGAs permitem ao projetista mapear os algoritmos diretamente em hardware, melhorando o seu desempenho, quando são

comparados com as suas respectivas implementações em software [4]. Uma questão relevante nos atuais projetos em sistemas embarcados baseados em FPGAs é a possibilidade de utilizar tanto o software como o hardware, usando as técnicas do conhecido co-projeto hardware/software (*Hardware/Software co-design*).

### **1.3 APRESENTAÇÃO DO TRABALHO.**

Com o propósito de conhecer e entender o problema e assim alcançar os objetivos propostos para esta pesquisa, o documento foi dividido nas seguintes partes: No capítulo 2 se apresenta uma revisão teórica dos temas centrais da pesquisa, *i.e.*, a soldagem GTAW, o sensoriamento infravermelho, a técnica de detecção de mudança e os dispositivos reconfiguráveis.

O capítulo 3 encontra-se as técnicas e a forma como foi encarado o problema, tanto para a parte do sistema embarcado quanto à parte experimental na bancada de soldagem.

O capítulo 4 relata os sucessos conquistados com os resultados da pesquisa. Apresenta tanto os resultados na área da soldagem como na área dos dispositivos reconfiguráveis que finalmente conseguem a detecção de descontinuidades para o processo de soldagem.

As conclusões e as perspectivas para trabalhos futuros são apresentadas no capítulo 5.

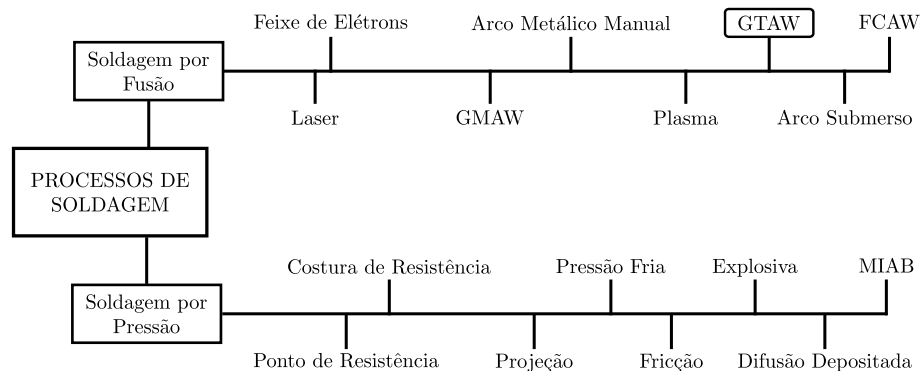
## 2 REVISÃO BIBLIOGRÁFICA

### 2.1 SOLDAGEM

Existem muitas definições para descrever o que é a soldagem, algumas falam da união só de peças metálicas enquanto outras afirmam que tem que haver um material de adição na junta de união. Considerando um bom senso entre todas as definições de soldagem tem-se a seguinte definição:

Processo de união de materiais utilizado para obter a coalescência (união) localizada de metais e não-metais, produzida por aquecimento até uma temperatura adequada, com ou sem a utilização de pressão e/ou material de adição [5].

Com base nesta definição, os processos de soldagem podem ser classificados em aqueles que se baseiam na aplicação de pressão e os que utilizam altas temperaturas para atingir a união [6]. Alguns dos processos mais importantes são mostrados na figura 2.1.



**Figura 2.1.** Classificação de alguns dos processos de soldagem mais importantes.

O processo de soldagem por pressão consiste na aplicação de pressões elevadas que deformam a superfície dos metais, permitindo que a distância média entre as superfícies a serem unidas seja reduzida conseguindo com que as forças interatômicas pudessem agir, formando ligações permanentes [7]. O aquecimento das peças a serem unidas pode ser usado para facilitar a deformação dos metais.

O processo de soldagem por fusão inclui a maioria dos processos mais versáteis empregados atualmente, e são usadas subclassificações para facilitar o estudo destes processos, dependendo da fonte de calor, o método de proteger a poça de fusão, e os modos de adição de metal de adição. A fonte de calor predominante é um arco elétrico produzido por corrente contínua (CC) ou fontes de corrente alternada (AC). Dentro dos processos de soldagem de arco, as classificações são baseadas no gás de proteção e/ou no metal de adição [7, 8]. Como se pode ver na figura 2.1, dentro dos processo de soldagem por fusão encontra-se o processo de Soldagem a Arco-Gás com eletrodo de Tungstênio (*Gas Tungsten Arc Welding* – GTAW), processo que foi o escolhido para este trabalho.

## 2.2 O PROCESSO DE SOLDAGEM GTAW

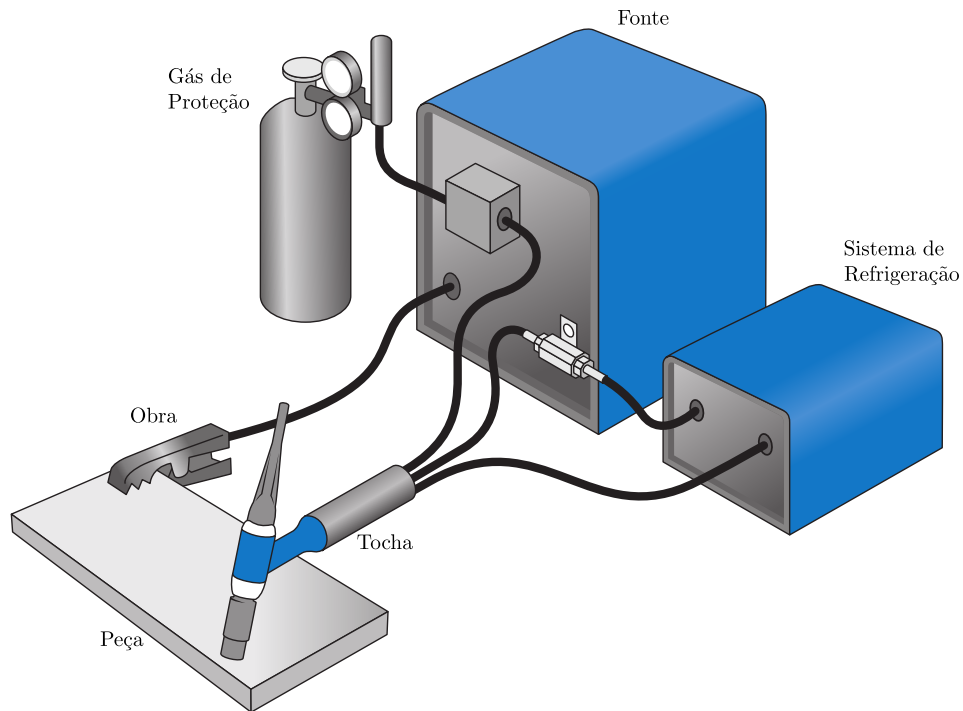
O processo de Soldagem a Arco-Gás com eletrodo de Tungstênio (*Gas Tungsten Arc Welding* – GTAW), ou TIG (*Tungsten Inert Gas*) como também é conhecido, faz parte dos processos de soldagem por fusão. Na soldagem GTAW o arco é produzido entre o eletrodo de tungstênio não consumível e a peça de trabalho [9]. A proteção do eletrodo e da zona da solda é feita por um gás inerte que é soprado pelo bocal da tocha, normalmente o argônio (Ar) ou uma mistura de gases inertes. Um Metal de adição pode ser ou não usado.

O equipamento básico num processo de soldagem GTAW consiste de uma fonte de energia, tocha com o eletrodo de tungstênio, gás de proteção e o sistema para a abertura do arco. A figura 2.2 apresenta o esquema essencial do processo GTAW manual.

A principal vantagem do processo GTAW em relação a outros processos de soldagem é que este solda uma maior gama de metais incluindo aço, níquel, titânio, alumínio, magnésio, cobre, bronze, incluso ouro.

A seguir se apresentam algumas vantagens e limitações do processo GTAW [10]:

- Excelente controle da poça de fusão.
- Permite soldagem sem o uso de metal de adição.
- Permite mecanização e automação do processo.
- Produz soldas de alta qualidade e excelente acabamento.
- Gera pouco ou nenhum respingo.
- Exige pouca ou nenhuma limpeza após a soldagem.
- Permite a soldagem em qualquer posição.



**Figura 2.2.** Esquema essencial do processo GTAW manual.

- Produtividade relativamente baixa.
- Custo de consumíveis e equipamento é relativamente elevado.

### 2.2.1 Algumas variáveis do processo.

A tensão do arco, a corrente de soldagem, a velocidade de deslocamento e o gás de proteção são as variáveis básicas no processo de soldagem GTAW, tendo todas estas variáveis uma forte interação entre elas.

Em relação à corrente de soldagem pode-se considerar, de forma geral, que ela controla proporcionalmente a penetração da solda. A corrente também afeta a tensão de arco, que é o nome dado à tensão entre o eletrodo e a peça. Para um comprimento de arco, um aumento na corrente produz um aumento na tensão de arco.

Estas são outras variáveis que influem no processo de soldagem [11]:

- Tipo de corrente (corrente contínua com o eletrodo positivo, CC+, corrente contínua com o eletrodo negativo, CC-, e corrente alternada, CA).
- Tensão de operação (ou de soldagem).
- Comprimento do arco.



- Velocidade de alimentação de metal de adição.
- Diâmetro do eletrodo.
- Distância do ponto de tomada de corrente à peça.
- Ângulo da tocha, ou eletrodo, em relação à peça.

## 2.2.2 Equipamento.

O equipamento usado no processo GTAW esta constituído principalmente por a *Fonte*, a *Tocha*, o *Eletrodo* e o *Gás de Proteção*.

### 2.2.2.1 Fontes.

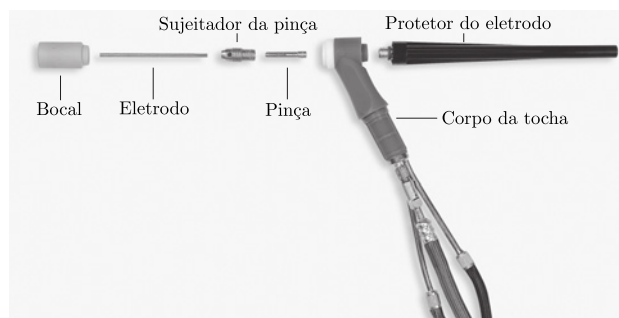
As fontes usadas para o processo GTAW apresentam recursos como o uso de corrente contínua, pulsada e de corrente alternada com onda retangular. A primeira técnica permite um melhor controle da poça de fusão na soldagem de peças de pequena espessura ou fora da posição plana, pois ao haver variação no comprimento do arco, haverá variação na tensão e a variação na corrente será mínima. A segunda é usada na soldagem de ligas de Al ou de Mg e dispensa a necessidade de se manter um ignitor de alta frequência operando continuamente [9, 10]. A fonte usada neste trabalho foi a *Inversal 450*.

### 2.2.2.2 Tochas.

As tochas, no processo GTAW são classificadas basicamente pelo seu mecanismo de refrigeração. As tochas refrigeradas a gás que são mantidas na temperatura adequada pelo efeito de resfriamento causado pelo próprio gás de proteção. Estas tochas estão limitadas a uma corrente máxima de cerca de 200 A. As tochas refrigeradas a água, normalmente em circuito fechado, como a usada neste trabalho, podem suportar correntes de até 1000 A, e são as mais empregadas em equipamentos automatizados de GTAW. A figura 2.3 apresenta uma imagem de uma tocha GTAW e as partes que a compõem.

### 2.2.2.3 Eletrodos.

Os eletrodos no processo GTAW são não consumíveis e também servem como um dos terminais do arco que irá gerar o calor para o processo. Ao aproximar-se da sua temperatura de fusão (3410 °C), o tungstênio torna-se termoiônico, como uma fonte disponível de elétrons. Ele



**Figura 2.3.** Tocha GTAW refrigerada por água.

alcança esta temperatura através de aquecimento por resistência e, caso não houvesse um forte efeito de resfriamento pela saída dos elétrons de sua extremidade, esta ponta poderia fundir-se.

O eletrodo usado neste trabalho foi o eletrodo classificado como EWTh-2, reconhecível pela cor vermelha em um dos seus extremos, o que indica que, além do tungstênio, o eletrodo tem um 2% de óxido de tório. O óxido de tório incrementa a emissividade termoiônica do tungstênio, permitindo a operação em correntes mais elevadas. Os eletrodos tórinados mantêm um fino perfil da ponta durante a soldagem, o que é desejável na soldagem de aços.

#### 2.2.2.4 Gás de proteção.

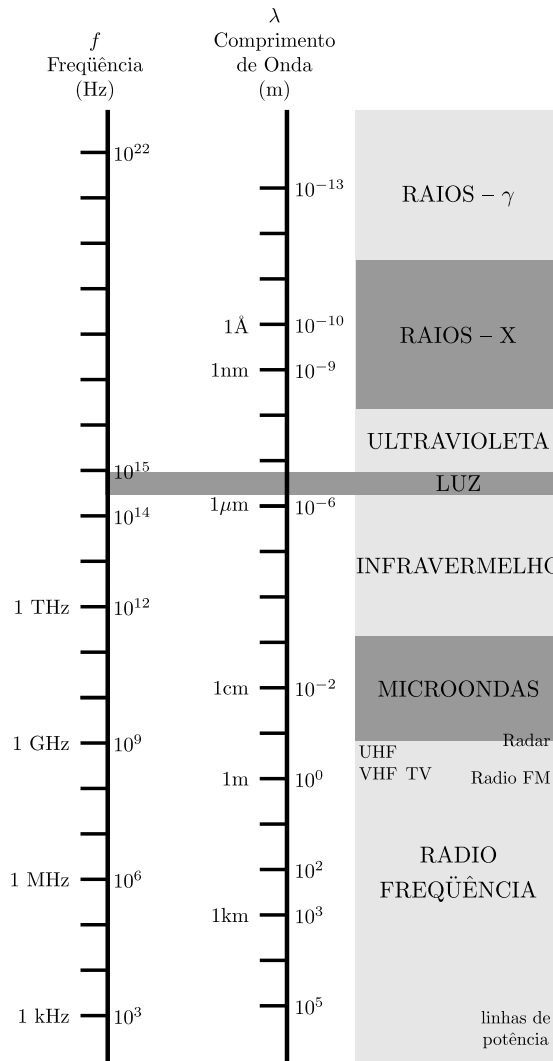
Todos os processos de soldagem a arco utilizam algum método para proteger a poça de fusão da atmosfera. Sem essa proteção, o metal fundido reage com os gases na atmosfera e produz porosidade no cordão de solda reduzindo a qualidade da solda.

Os dois gases mais utilizados no processo GTAW são o Argônio e o Hélio. Os gases de proteção em aplicações de soldagem devem ser de alta pureza (99,995%). O tipo de gás e a sua vazão durante a soldagem têm influência nas características e o comportamento do arco e do cordão resultante. A densidade do gás é outro fator que influencia no resultado da solda.

## 2.3 SENSORIAMENTO INFRAVERMELHO

### 2.3.1 Radiação infravermelha

A radiação eletromagnética abrange uma ampla faixa de comprimentos de onda e frequências, como por exemplo as ondas de radio, as microondas, o infravermelho e a luz visível. A figura 2.4 apresenta as principais seções em que o espectro eletromagnético é usualmente dividido [12, 13]. Para o propósito deste trabalho se fará referencia apenas à faixa do infravermelho.



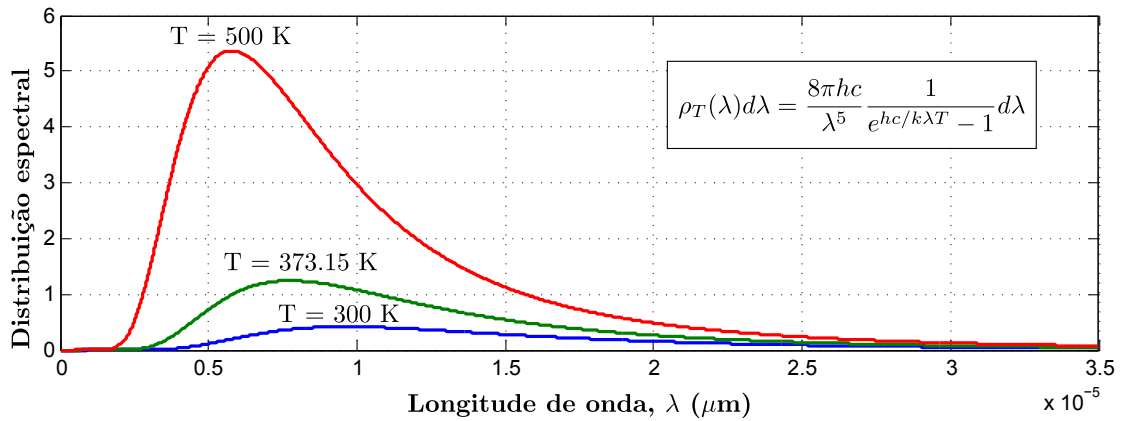
**Figura 2.4.** Esquema do espectro eletromagnético.

A faixa do infravermelho, se estende aproximadamente desde  $3 \times 10^{11}$  Hz até  $4 \times 10^{14}$  Hz. Como o seu nome o indica, esta região do espectro se encontra justo abaixo da luz vermelha. O infravermelho é às vezes subdividido em quatro regiões: o infravermelho próximo, 780 – 3000 nm (perto do visível); o infravermelho médio, 3000 – 6000 nm; o infravermelho longo 6000 – 15000 nm; e o infravermelho extremo 15000 nm – 1.0 mm.

As moléculas de qualquer objeto a uma temperatura acima do zero absoluto ( $-273^\circ\text{C}$ ) emitirão radiação infravermelha aleatoriamente em todas as direções, cuja intensidade vai depender da temperatura e das propriedades do material.

A potência emissiva, definida como o fluxo de calor radiado de uma superfície, é distribuída sobre uma ampla faixa de comprimentos de onda para um objeto a uma temperatura  $T$ . A potência emissiva de um corpo negro, descrita pela lei de Planck [14] é representada na figura 2.5 para diferentes temperaturas. Para temperaturas na faixa da temperatura ambiente até aquelas atingidas em um processo de soldagem, o pico para a potência emissiva acontece em uma

longitude de onda na faixa do infravermelho meio até o espectro visível [15].



**Figura 2.5.** Distribuição espectral de Planck para um corpo negro. Têm plotadas as temperaturas para a temperatura ambiente (300 K) e a temperatura de ebulição da água (373.15 K).

Uma superfície real difere de um corpo negro em que esta tem uma emissividade menor que 1 e pode refletir e/ou transmitir a radiação eletromagnética incidente. As condições da superfície são afetadas pelo grau de rugosidade e a presença de agentes externos como óxidos, óleos, água e poeira.

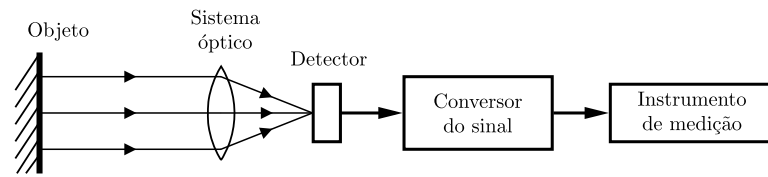
### 2.3.2 Medição da temperatura por radiação

A medição da temperatura por radiação ou o sensoriamento da radiação infravermelha é uma técnica para medir temperatura que é amplamente usada em áreas como o monitoramento ambiental, astronomia, investigações militares, inclusive na vida diária. A energia infravermelha radiada é geralmente medida com dispositivos que respondem ao calor gerado na absorção do infravermelho por uma superfície obscura, estes dispositivos são os pirômetros [15]. Este trabalho utiliza o sensoriamento infravermelho para medir as variações da radiação infravermelha do arco, no processo de soldagem GTAW, que possam ser produzidas por descontinuidade no cordão de solda.

Os pirômetros, também conhecidos como termômetros infravermelhos, ou termômetros de radiação, são termômetros de não contato, os quais medem a temperatura de um corpo baseados na sua radiação térmica emitida. Os sensores infravermelhos que operam na banda de 0,7-20  $\mu\text{m}$  são normalmente utilizados para a medição prática. Para comprimentos de onda superiores a 20  $\mu\text{m}$ , a energia da radiação incidente é demasiado pequena para ser detectada por estes [15].

Os pirômetros podem classificar-se como [16, 17]: (i) pirômetros operados manualmente, onde o olho do operador faz as vezes de comparador, entre o sinal de radiação que vem da fonte, com um sinal de referência. (ii) pirômetros operados automaticamente, na figura 2.6 apresenta-se

o diagrama de blocos de um pirômetro operado automaticamente, e que está composto por as seguintes partes principais:



**Figura 2.6.** Diagrama de blocos de um pirômetro automático.

- Sistema óptico que concentra e enfoca no detector a radiação incidente emitida por um corpo.
- Detector de radiação infravermelha que pode ser um sensor térmico ou fotoelétrico.
- Conversor do sinal, acondiciona o sinal de saída do detector.
- Instrumento de medição.

### 2.3.2.1 Detectores de radiação térmica

O detector de radiação térmica é a parte principal de um sistema de sensoriamento infravermelho. Pode-se classificar em dois tipos segundo sua capacidade de responder à radiação térmica. O primeiro tipo são os *Detectores Térmicos* ou de *Radiação* que é o tipo de detector usado pelo sensor que se vai usar neste trabalho. O segundo tipo são os *Detectores Fotônicos* ou *Quânticos* [16, 17].

Nos *Detectores Térmicos*, seu sinal de saída é proporcional ao calor resultante da absorção da radiação infravermelha pela superfície do elemento sensor, enquanto que o sinal de saída dos *Detectores Fotônicos* está relacionado com a excitação gerada pelos fótons incidentes

## 2.4 O FILTRO KALMAN

Teoricamente, o filtro Kalman é um estimador para o que é chamado *Problema Quadrático Linear*, que é o problema de estimar o estado instantâneo de um sistema linear perturbado por um ruído branco [18].

Para fazer uso do filtro Kalman se requer de um modelo de espaço de estados em tempo discreto que descreva a dinâmica do sinal [19, 20], como é apresentado nas seguintes equações de

diferenças,

$$\begin{aligned}x_k &= F_{k-1}x_{k-1} + G_{k-1}u_{k-1} + w_{k-1} \\y_k &= H_kx_k + v_k\end{aligned}\tag{2.1}$$

em que  $y_k$  é o sinal medido;  $F$ ,  $G$  e  $H$  são termos conhecidos que descrevem o processo e  $x_k$  é o de estados desconhecido.  $F$  é o termo de relação entre o estado anterior  $k - 1$  e o estado atual  $k$ , o termo  $G$  relaciona a entrada opcional  $u$  com o estado  $x$ ,  $H$  é o termo de relação entre o estado e a medida  $y_k$ . O ruído do processo é  $\{w_k\}$ , e  $\{v_k\}$  é o ruído da medição, estes são não-correlacionados, com distribuição normal, média zero variâncias conhecidas  $Q_k$  e  $R_k$ , respectivamente, a equação (2.2) descreve isto,

$$\begin{aligned}w_k &\sim \mathcal{N}(0, Q_k) \\v_k &\sim \mathcal{N}(0, R_k) \\E[w_k w_j^T] &= Q_k \delta_{k-j} \\E[v_k v_j^T] &= R_k \delta_{k-j} \\E[v_k w_j^T] &= 0\end{aligned}\tag{2.2}$$

$\delta_{k-j} = 1$  se  $k = j$ , e  $\delta_{k-j} = 0$  se  $k \neq j$ .

Na pratica, a variância do ruído do processo  $Q$  e a variância do ruído da medida  $R$  podem mudar para cada medição, embora para este trabalho assume-se que elas são constantes.

O objetivo do filtro é estimar o estado  $x_k$  baseado no conhecimento do estado anterior do sistema e a medição  $y_k$ . Se tiver-se disponíveis todas as medidas antes do tempo  $k$ , se tem uma estimativa *a priori*, denotada por  $\hat{x}_{k|k-1}$ :

$$\hat{x}_{k|k-1} = E[x_k | y_1, y_2, \dots, y_{k-1}]\tag{2.3}$$

Se tiver-se disponíveis todas as medidas incluindo a feita no tempo  $k$  para fazer a estimativa de  $x_k$ , que é representada por  $\hat{x}_k$ , se tem uma estimativa *a posteriori*, denotada por  $\hat{x}_{k|k+1}$  e expressada da seguinte maneira:

$$\hat{x}_{k|k+1} = E[x_k | y_1, y_2, \dots, y_k]\tag{2.4}$$

Tanto  $\hat{x}_{k|k-1}$  como  $\hat{x}_{k|k+1}$ , são estimativas da mesma quantidade,  $x_k$ . Embora,  $\hat{x}_{k|k-1}$  é a estimativa antes de considerar a medição  $y_k$ , e  $\hat{x}_{k|k+1}$  depois de ter a medição  $y_k$ .

O termo usado para representar a variância do erro estimado é  $P_k$ .  $P_{k|k-1}$  representa a variância do erro estimado *a priori* e  $P_{k|k+1}$  representa a variância do erro estimado *a posteriori*, segundo as equações (2.5) e (2.6), respectivamente.

$$P_{k|k-1} = E[(x_k - \hat{x}_{k|k-1})(x_k - \hat{x}_{k|k-1})^T]\tag{2.5}$$

$$P_{k|k+1} = E \left[ (x_k - \hat{x}_{k|k+1}) (x_k - \hat{x}_{k|k+1})^T \right] \quad (2.6)$$

## 2.4.1 O algoritmo do filtro Kalman

O algoritmo do filtro Kalman pode-se explicar como uma forma de controle realimentado [21]: primeiro o filtro estima o estado de um processo em algum tempo  $k$  e então obtém uma realimentação em forma de uma “medida ruidosa”. Desta maneira, as equações do filtro Kalman podem-se dividir em dois grupos: equações para a *atualização do tempo* e as equações para a *atualização da medida*. As equações para a *atualização do tempo* projetam no tempo o estado atual e a variância do erro para obter a estimativa *a priori* para o seguinte estado  $k + 1$ . As equações para a *atualização da medida* são responsáveis da realimentação, ou seja, incorporam a nova medição e a estimativa *a priori* para obter uma estimativa *a posteriori* melhorada. A seguir são apresentadas as equações para a *atualização do tempo* e *atualização da medida* no algoritmo do filtro Kalman.

### 2.4.1.1 Atualização do tempo

A estimação inicial de  $x_0$  antes que qualquer medição esteja disponível é:  $\hat{x}_{0|-1}$ . Considerando que ainda não se têm medidas disponíveis para estimar  $x_0$ , toma-se  $\hat{x}_{0|-1}$  como o valor esperado do estado inicial  $x_0$ :

$$\hat{x}_{0|-1} = E(x_0) \quad (2.7)$$

As estimativas do processo iniciam-se com  $\hat{x}_{0|-1}$ . Para estimar  $\hat{x}_{1|0}$ , usa-se a equação para a propagação da média de  $x$  no tempo [20]:  $\bar{x}_k = F_{k-1}\bar{x}_{k-1} + G_{k-1}u_{k-1}$ . Então para obter  $\hat{x}_{1|0}$  de  $\hat{x}_{0|-1}$ , se tem:

$$\hat{x}_{1|0} = F_0\hat{x}_{0|-1} + G_0u_0 \quad (2.8)$$

De forma geral se tem:

$$\hat{x}_{k|k-1} = F_{k-1}\hat{x}_{k-1|k} + G_{k-1}u_{k-1} \quad (2.9)$$

A equação (2.9) é chamada de equação de atualização do tempo para  $\hat{x}$ . Para o tempo  $k - 1$  até o tempo  $k$ .

Agora, para obter a equação de atualização do tempo para  $P$ , se inicia com  $P_{0|-1}$ , que é a variância da estimativa inicial de  $x_0$ . Se o estado inicial é perfeitamente conhecido, então  $P_{0|-1} = 0$ , caso contrário  $P_{0|-1} = \infty I$ . O termo  $P_{0|-1} = 0$  representa a incerteza da estimativa

inicial de  $x_0$ :

$$\begin{aligned} P_{0|-1} &= E \left[ (x_0 - \bar{x}_0) (x_0 - \bar{x}_0)^T \right] \\ &= E \left[ (x_0 - \hat{x}_0) (x_0 - \hat{x}_0)^T \right] \end{aligned} \quad (2.10)$$

Da mesma maneira que com a média, com a variância usa-se a propagação desta no tempo [20]:  $P_k = F_{k-1}P_{k-1}F_{k-1}^T + Q_{k-1}$ , para calcular  $P_{1|0}$  a partir de  $P_{0|-1}$ :

$$P_{1|0} = F_0 P_{0|-1} F_0^T + Q_0 \quad (2.11)$$

De forma geral se tem:

$$P_{k|k-1} = F_{k-1} P_{k-1|k-1} F_{k-1}^T + Q_{k-1} \quad (2.12)$$

Esta é a equação para a atualização do tempo de  $P$ . O filtro Kalman trabalha propagando a média e a variância do estado através do tempo.

As equações (2.9) e (2.12), são as equações para a atualização do tempo do filtro Kalman que acontece entre os tempos  $k - 1$  e  $k$ .

### 2.4.1.2 Atualização da medida

Para a atualização da medida [20, 21], começa-se encontrando uma equação que calcule uma estimativa do estado *a posteriori*  $x_{k|k+1}$  como uma combinação linear do estado *a priori*  $x_{k|k-1}$  e uma diferença ponderada entre a medição atual  $y_k$  e uma predição na medida  $Hx_{k|k-1}$  como se apresenta na equação (2.13).

$$\hat{x}_{k|k+1} = \hat{x}_{k|k-1} + K_k (y_k - H_k \hat{x}_{k|k-1}) \quad (2.13)$$

A diferença  $(y_k - H_k \hat{x}_{k|k-1})$  na equação (2.13) é chamada de *inovação* da medida, ou *resíduo*. O resíduo reflete a discrepância entre a predição da medida  $H_k \hat{x}_{k|k-1}$  e a medição atual  $y_k$ . Um resíduo de zero indica que os dois coincidem no valor.

A matriz  $K$  em (2.13) é o ganho de Kalman que minimiza a variância do erro *a posteriori*, e pode-se expressar como

$$K_k = P_{k|k-1} H_k^T (H_k P_{k|k-1} H_k^T + R_k)^{-1} \quad (2.14)$$

A variância do erro estimado a posteriori se apresenta na equação (2.15)

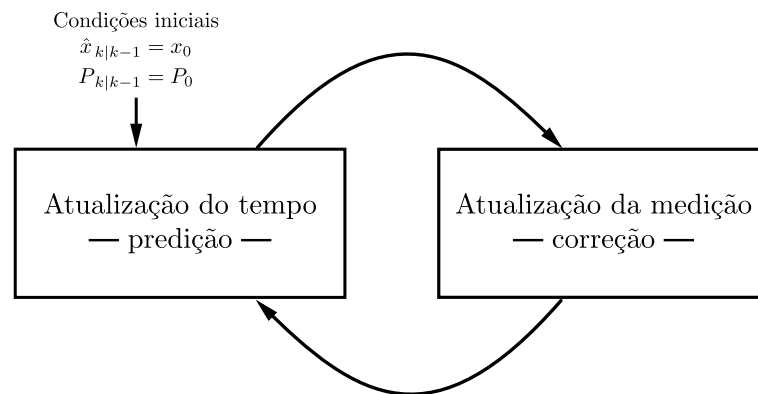
$$P_{k|k+1} = (I - K_k H_k) P_{k|k-1} \quad (2.15)$$



Das equação (2.14) e (2.15) pode-se olhar que o calculo de  $K_k$  e  $P_{k|k+1}$  não dependem da medida  $y_k$ . Isto implica que o ganho de Kalman  $K_k$  pode se calcular *off-line* e ser usado no momento em que a medição é feita.

As equações (2.13), (2.14) e (2.15), são as equações para a atualização da medida do filtro Kalman que acontece no tempo  $k$ .

As equações de *atualização do tempo* também podem ser pensadas como equações de *predição*, enquanto as equações de *atualização da medida* podem ser pensadas como equações de *correção*. O fim, na figura 2.7 é esquematizado o algoritmo do filtro Kalman.



**Figura 2.7.** Esquema geral do algoritmo do filtro Kalman .

## 2.4.2 Parâmetros e ajuste do filtro

Na implementação do filtro, a variância do ruído da medição  $R$  é normalmente medida antes da operação do filtro. Medir  $R$  é geralmente possível, porque se tem capacidade de medir o processo de qualquer maneira, assim podem-se tomar algumas medidas de maneira *off-line* a fim de determinar a variância do ruído medido.

A determinação da variância do ruído do processo  $Q$  é geralmente mais difícil, a maioria das vezes o processo que se esta estimando não pode ser observado diretamente. Normalmente este valor é suposto baixo [21].

O desempenho do filtro pode ser melhorado ajustando os parâmetros. Este ajuste é normalmente realizado *off-line*.

## 2.5 DETECÇÃO DE MUDANÇA

A base suposta na estimação do sinal consiste em que a medição  $y_k$  tem uma componente determinística  $x_k$ , e um ruído branco de natureza aditivo  $e_k$ ,

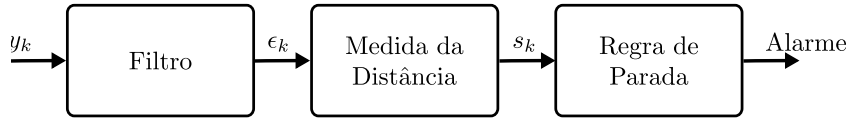
$$y_k = x_k + v_k \quad (2.16)$$

A determinação de  $x_k$  de  $y_k$  é a *estimação*, e encontrar uma mudança rápida ou abrupta em  $x_k$  é a *deteccção de mudança* ou *alarma*, que inicia no *tempo de mudança*  $k$ .

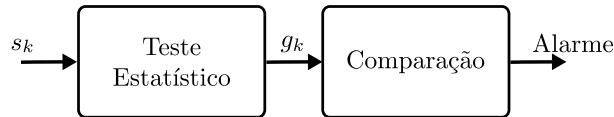
É assumido para o modelo que [19]:

- A componente determinística  $x_k$  é submetida a uma mudança abrupta no tempo  $k$ . Uma vez a mudança é detectada, o processo de vigilância inicia de novo para detectar a próxima mudança.
- Em uma aproximação estatística, se assumirá que o ruído é branco e tem uma distribuição normal:  $v_k \sim \mathcal{N}(0, R)$ .

O algoritmo para a deteção de mudança é representado na figura 2.8 [19].



(a) Fluxograma da deteção de mudança baseado no resultado do filtro.



(b) Regra de parada composta por o teste estatístico e a comparação.

**Figura 2.8.** Passos do modelo geral para um processo de deteção de mudança.

O bloco de filtro faz referência ao filtro Kalman exposto anteriormente, os demais blocos são descritos a seguir.

### 2.5.1 Medida da distância

Existem muitas possibilidades para medir a distância  $s_k$ . A que foi usada neste trabalho toma os resíduos, que é a diferença entre o dado lido e o dado estimado,

$$s_k = \epsilon_k = y_k - \hat{x}_{k-1} \quad (2.17)$$

em que  $\hat{x}_{k-1}$  é o valor estimado pelo filtro Kalman. Este é apropriado para a mudança da média, e que tem que ser robusto para mudanças da variância.

O resultado do cálculo da distância é a entrada para a regra de parada.

## 2.5.2 Regra de parada com teste CUSUM

A regra de parada é usada em supervisão para dar um alarme quando  $x_k$  ultrapassado um limiar  $h$ .

Dentro da regra de parada se encontra o *teste estatístico* auxiliar  $g_k$  e a *comparação*.

O teste estatístico é um teste de hipóteses. Quando existe um particular interesse em decidir sobre a validade ou não de uma hipótese específica, o teste de hipóteses dá a estrutura para fazer isso. Neste contexto se assume uma hipótese, denominada hipótese nula, dada por  $H_0$  e se apresenta uma hipótese alternativa, dada por  $H_1$ . Essa será assumida caso  $H_0$  falhe.

No caso, a hipótese  $H_0$  diz que não há defeito no sinal lido pelo sensor. Logo, assumir  $H_1$  significa que houve defeito ou mudança, assim, há probabilidade da presença de uma descontinuidade no sinal.

Baseado no cálculo da distância, é avaliado se o valor estimado  $\hat{x}_{k-1}$  está dentro de um intervalo aceito por  $y_k$  lido, ou seja, se aquele faz parte da distribuição deste, ou se está nos limites de dispersão estabelecidos. Um valor pequeno de distância significa que a diferença entre aqueles dois valores não é apreciável, então se aceita  $H_0$ . Mas se o valor for grande, pode ultrapassar os limites estabelecidos pelo padrão e a divergência será notável, então se rejeita  $H_0$  e se aceita  $H_1$ .

Um dos algoritmos mais utilizados nesta etapa é o de somas cumulativas ou CUSUM (Cumulative SUM) [19]:

$$g_k = \max(g_{k-1} + s_k - \nu, 0) \quad (2.18)$$

A comparação tem a priori a informação de quão grande é o limiar  $h$  e quando  $g_k$  ultrapassa  $h$  o alarme é ativado,

$$\text{Alarme se: } g_k > h \quad (2.19)$$

Se  $g_k > h$  :  $g_k = 0$  e  $k = k_a$ . O termo  $k_a$  é o tempo em que o alarme aconteceu. Para prevenir acumulações positivas, que levem para um alarme falso, o parâmetro de deslocamento  $\nu$  é subtraído em cada iteração. Para prevenir as acumulações negativas, que podem aumentar o tempo de detecção depois de acontecer alguma mudança, o teste estatístico restabelece,  $g_k = 0$ , para a seguinte iteração. A magnitude do parâmetro de deslocamento  $\nu$  no instante  $k$  pode ser definida como  $\nu \triangleq x_{k+1} - x_k$ .

## 2.6 TECNOLOGIA DOS FPGAs

Um arranjo de portas programáveis no campo ou FPGA (*Field Programmable Gate Array*) é um dispositivo lógico que contém um arranjo bi-dimensional de células lógicas genéricas e interruptores. Os FPGAs podem ser incluídos em uma classe de dispositivos chamados lógicos programáveis em campo, ou *Field-Programmable Logic* (FPL). Os FPL são definidos como dispositivos programáveis que contêm de maneira repetida campos de pequenos blocos e elementos, chamados, no caso da Xilinx, blocos lógicos configuráveis (*Configurable Logic Block* – CLB) [22].

Um bloco lógico usualmente contém circuitos combinacionais configuráveis com flip-flop D (D-FF). O método mais comum para implementar circuitos combinacionais configuráveis é a *Look-Up Table* (LUT). Uma LUT de  $n$  entradas poder ser considerada como uma memória de  $2^n \times 1$ . Escrevendo corretamente o conteúdo da memória, é possível implementar qualquer função combinacional de  $n$  entradas [23].

O tamanho dos blocos lógicos esta relacionado com o termo *granularidade* do dispositivo, este termo faz referência ao esforço requerido para completar o roteamento entre os blocos lógicos. Em geral existem três tipos de granularidade, **Granularidade fina** (mar de portas), **Granularidade média** (FPGA) e **Granularidade grossa** (*Complex Programmable Logic Device* – CPLD) [22].

**Granularidade fina.** A célula lógica básica consiste de uma única comporta NAND e um latch. Cada célula pode ser usada para implementar uma única função, por exemplo uma portas lógica primitiva ou elemento de memória.

**Granularidade média.** As células lógicas básicas são tipicamente tabelas ou multiplexadores lógicos dedicados. Blocos programáveis de entrada/saída com flip-flops são unidos fisicamente nas fronteiras do dispositivo.

**Granularidade grossa.** Estes dispositivos têm uma combinação dos assim chamados Dispositivo Lógico Programável Simples (*Simple Programmable Logic Device* – SPLD). Os SPLD são Arranjos Lógicos Programáveis (*Programmable Logic Array* – PLA) implementado como um arranjo de AND/OR e blocos universais de entrada/saída.

Uma consideração importante respeito da granularidade é que as implementações de granularidade fina requerem um número relativamente grande de conexões de entrada e saída em cada bloco, em relação à quantidade de funcionalidade que pode ser suportada por esses blocos. Quando a granularidade é maior, a quantidade de conexões dos blocos diminui em comparação com a quantidade de funcionalidade suportada. Isto é importante porque a quantidade de

interconexões programáveis é responsável dos atrasos dos sinais através do FPGA.

### 2.6.1 Aplicações

Quando os FPGAs apareceram pela primeira vez nos anos 80, foram amplamente utilizadas para implementar lógica básica, máquinas de estado de complexidade média, e um limitado processamento de dados. Durante os anos 90 o tamanho e a complexidade dos FPGAs começou a aumentar e sua aplicação fundamental foi as telecomunicações [24]. Para finais dos anos 90 os FPGAs eram também utilizados em automóveis e aplicações industriais.

Os FPGAs são utilizadas regularmente para o prototipagem de ASIC ou para prover uma plataforma de hardware para verificar implementações físicas de novos algoritmos. Por outro lado, seu baixo custo de desenvolvimento e curto tempo para estar no mercado fazem que os FPGAs gradualmente sejam mais usados.

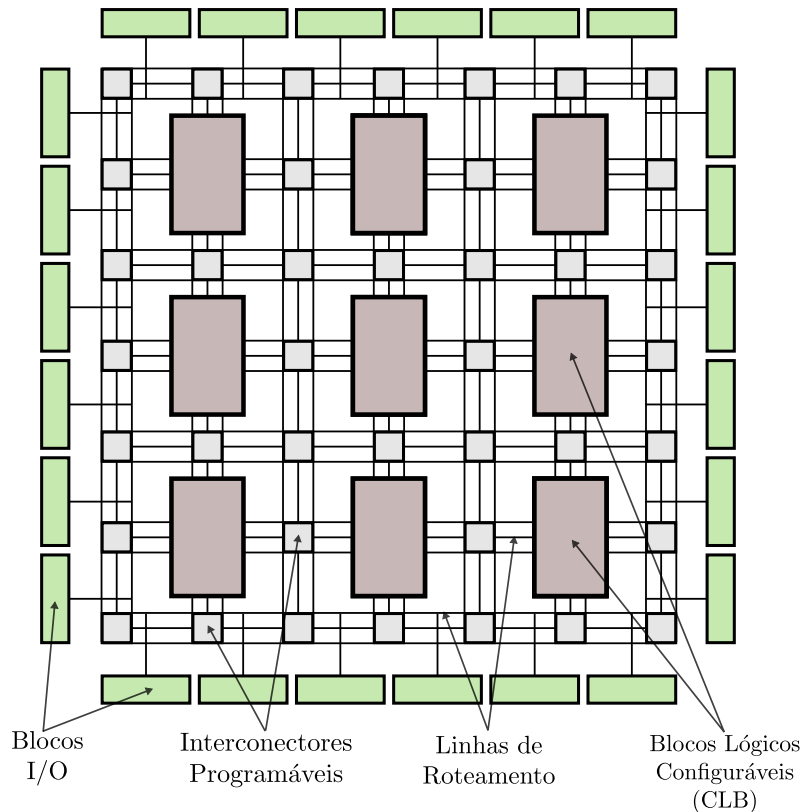
FPGAs de alto rendimento que contem milhões de portas lógicas podem se encontrar no mercado. Alguns destes dispositivos incluem microprocessadores, entradas e saídas de alta velocidade, e algumas outras características. Como resultado disso, os FPGAs são usados atualmente em várias tarefas, que incluem comunicações, processamento de imagens e outras aplicações de processamento de sinais; tarefas que podem ser parte dos componentes de *Systems on Chip* (SoC) que contém tanto elementos hardware quanto de software [24].

### 2.6.2 Arquitetura dos FPGAs

A arquitetura mais comum de um FPGA é apresentado na figura 2.9. Um bloco lógico pode ser configurado para realizar uma simples função, e um interconector programável pode ser personalizado para prover interconexões dentro do bloco lógico, são estes elementos os que fazem a FPGA reconfigurável. Um desenho personalizado pode ser implementado especificando a função de cada bloco lógico e ajustando seletivamente as conexões dos interconectores. Uma vez que o desenho e a síntese sejam completados, pode-se gerar o *bitstream* de configuração e descarregá-lo no FPGA para obter o circuito personalizado.

Dentro dos elementos gerais que compõem um FPGA funcional (independente de seu fabricante) encontram-se [24, 25]:

- Blocos ou células lógicas, compostos de LUT e FF.
- Gerenciadores de relógio.
- Portas de entrada e saída para propósito geral.



**Figura 2.9.** Estrutura mais comum da arquitetura de um FPGA.

- Memórias embebidas.
- Multiplicadores embarcados.
- Processadores embarcados.
- Hardware de Propriedade Intelectual (*Intellectual Property* – IP).

Considerando o enfoque deste trabalho vai-se fazer uma descrição dos processadores embarcados e o hardware de propriedade intelectual (IP).

### 2.6.2.1 Processadores embarcados

Quase qualquer parte de um projeto eletrônico pode ser realizado em hardware ou software. Um dos principais pontos de partida do projetista é quão rápido se deseja que o projeto eletrônico leve a cabo suas tarefas. Nos FPGAs atuais tornaram-se disponíveis um ou mais microprocessadores embarcados. Isso fornece uma série de vantagens, não unicamente na diminuição de custo por não ter que usar um dispositivo a mais, mas também elimina um grande número de faixas e caminhos e pinos na placa do circuito, o que faz a placa menor e mais leve, além da flexibilidade

de misturar software e hardware em um chip. Estas são outras vantagens de usar processadores embarcados em FPGAs [26]:

- Capacidade para implementar todas ou a maioria das funcionalidades do sistema em um único dispositivo.
- Capacidade para implementar uma solução de processamento embarcado adaptável.
- Capacidade para implementar apenas a funcionalidade específica requerida.
- Capacidade para implementar uma solução de processamento escalável.
- O potencial para um melhor desempenho do sistema.
- Capacidade de suportar modificações no projeto durante o projeto.
- Otimização de interfaces processador-periférico.
- Otimização de implementações funcionais hardware versus software.
- Melhora a interação entre o projeto hardware e software.
- Mais eficiência na interface do sistema.
- possibilidade de utilizar o mesmo hardware para múltiplas aplicações.
- Potencial redução no custo de implementação.
- Capacidade de implementar coprocessadores personalizado.
- Capacidade para implementar aplicações tipo multiprocessador.

Neste contexto existem os processadores *Hard* que são microprocessador implantados fisicamente dentro do FPGA no momento de sua fabricação, e os processadores *Soft* que são microprocessadores criados dentro do FPGA usando os recursos dele. A seguir só se apresenta uma descrição dos microprocessadores *Soft*.

**Processadores *Soft*.** Dentro dos FPGAs é possível configurar um grupo de blocos lógicos programáveis para atuar como um ou mais processadores [24]. Estes são tipicamente chamados de *soft core*, mas podem ser mais precisamente categorizados como “*soft*” ou “*firm*” dependendo do jeito que a funcionalidade do microprocessador tinha sido mapeada dentro dos blocos lógicos. Os *Soft* cores são mais simples e lentos do que os *Hard* cores. O *Soft* core usado neste trabalho foi o processador *MicroBlaze* da Xilinx [27]. A seguir se apresentam algumas das vantagens e desvantagens dos processador *Soft*.

Vantagens oferecidas por um processador *Soft*:

- Bem otimizado para a arquitetura orientada.
- Relativamente fácil de modificar.
- Bom desempenho, utilização de recursos e consumo de potencia geralmente bem balanceado.
- Boa relação funcionalidade/desempenho.
- verificados em campo.
- Fácil de testar no ambiente.
- Acesso a certo nível de simulação.
- Alguma documentação disponível.
- Potencial aceso ao conhecimento do projeto.

Desvantagem que pode ter um processador *Soft*:

- Pouco incentivo para fazê-los altamente portáteis.
- Nível de documentação, configurabilidade, suporte para simulação e aceso ao projeto original pode ser limitado.
- O aceso a uma assistência avançada pode depender de condições negociáveis.

### 2.6.2.2 Processador *MicroBlaze*.

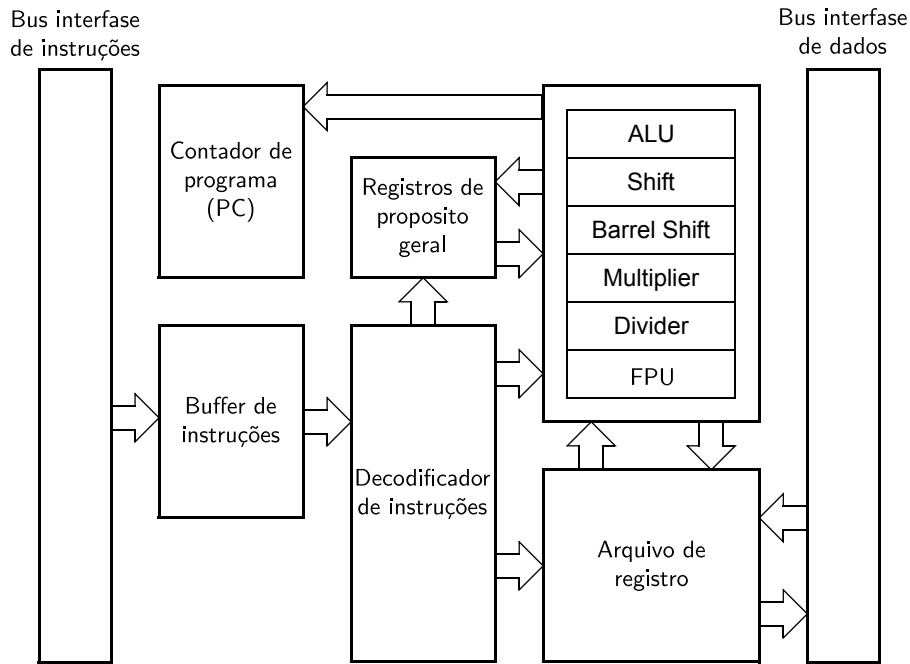
O processador *soft* embarcado *MicroBlaze* é um processador de 32 bit com um conjunto reduzido de instruções (RISC) otimizado para ser usado em FPGAs da Xilinx. A figura 2.10 mostra o diagrama de blocos funcional do core do *MicroBlaze* [27].

O processador *soft* *MicroBlaze* é altamente configurável, permitindo selecionar um conjunto de características fixas a critério do projetista, nas que se inclui:

- Registros de propósito geral de 32 bit.
- Palavras de instruções de 32 bit.
- Barramento de endereços de 32 bit
- Single issue pipeline

Uma descrição mais detalhada do processador *MicroBlaze* pode ser encontrada em [27].





**Figura 2.10.** Diagrama de blocos do core do *MicroBlaze*.

### 2.6.2.3 Blocos de propriedade intelectual (IP)

Os projetos desenvolvidos em FPGAs podem ser tão grandes e complexos que para a realização destes não seria viável criar cada parte dele desde zero. Uma solução é a reutilização de blocos funcionais para a realização das coisas “cotidianas” e investir mais recursos e tempo na realização daquelas coisas que definem o projeto mesmo. Estes blocos funcionais são tipicamente chamados, blocos de Propriedade Intelectual ou IP (*Intellectual Property*).

Cada distribuidor de FPGAs oferece sua própria seleção de *hard*, *firm* e *soft* IP. Os *Hard* IP são blocos implantados durante a fabricação do FPGA. Estes blocos são projetados para ser o mais eficientes em termos das características de cada família de FPGAs. Os *Soft* IP são bibliotecas contendo funções que podem ser incluídas nos projetos dos usuários. Estas funções são tipicamente feitas com uma linguagem de descrição de hardware. É comum que o IP seja também chamado de bibliotecas parametrizada ou LPM (*Library Parameterized Modules*), megafunções, macros, macros relacionamente alocados ou RPM (*Relationally Placed Macros*), cores, e cores sintetizáveis.

## 3 PROCEDIMENTO E METODOLOGIA

Os procedimentos e os métodos desenvolvidos são apresentados particularmente em duas partes. A primeira, chamada de *Sistema de detecção*, contém a descrição do co-projeto hardware/software desenvolvido ao redor do FPGA. A segunda, chamada de *Sistema de soldagem*, contém uma descrição dos componentes da bancada de soldagem e como ela foi utilizada. Adicionalmente, é feita uma descrição da maneira como as provas experimentais para a validação do sistema de detecção de descontinuidades foram elaboradas.

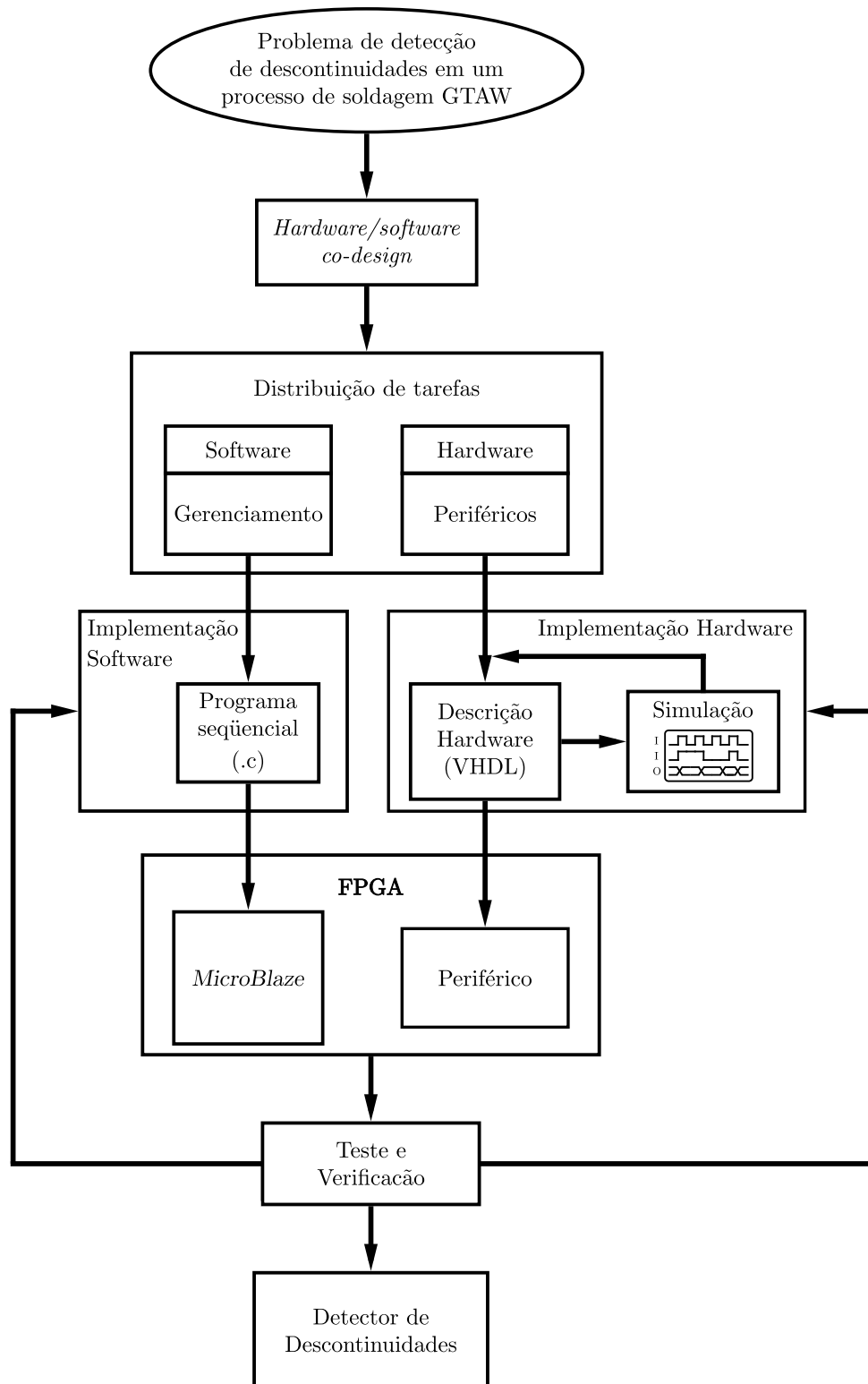
### 3.1 SISTEMA DE DETECÇÃO

O sistema de detecção representa uma abordagem que agrega ferramentas hardware e software, com o propósito de resolver o problema de detectar descontinuidades no processo de soldagem GTAW. O procedimento geral que foi usado está representado por um diagrama de blocos na figura 3.1. Todo este desenvolvimento hardware e software centra-se no FPGA, como é exposto a seguir.

#### 3.1.1 Co-projeto Hardware/Software

Um aspecto importante nos projetos de sistemas embarcados baseados em FPGAs é a possibilidade que eles fornecem de misturar as vantagens do hardware e o software, conhecida como técnica de co-projeto hardware/software (*Hardware/Software co-design*).

A técnica de co-projeto hardware/software é a distribuição de tarefas de uma aplicação entre um microprocessador que roda um software sequencialmente e um ou mais componentes de hardware personalizados chamados co-processadores ou periféricos, melhorando desta maneira o desempenho da aplicação em termos de tempo de projeto de hardware, tamanho, consumo de potência, frequência, etc. O projetista tem a vantagem de decidir que porção da aplicação será a parte seqüencial e implementá-la usando uma linguagem de programação de alto nível como C, além de decidir qual será a parte concorrente e descrevê-la utilizando uma linguagem de descrição hardware como VHDL o Verilog [28].



**Figura 3.1.** Diagrama de blocos do procedimento usado para desenvolver o sistema de detecção de descontinuidades com o FPGA.

### 3.1.2 Distribuição de tarefas

A distribuição de tarefas foi atribuída pela complexidade e prioridade de cada tarefa, além de considerar que, com a descrição hardware do sistema de detecção de mudança, aproximará

mais do processo *on-line* de detecção das descontinuidades em um processo de solda.

### 3.1.2.1 Porção sequencial

À porção sequencial que roda no processador *soft MicroBlaze* foi dada a tarefa do gerenciamento de todo o sistema, isto é: a interface de usuário e a comunicação e a transferência de dados entre alguns dos periféricos instanciados.

### 3.1.2.2 Porção concorrente

Considerando a prioridade do projeto, à porção concorrente foi conferida a descrição em hardware do algoritmo para a detecção das descontinuidades em soldagem. Além disso, esta divisão permite estudar o desempenho deste tipo de algoritmo de detecção de mudança em hardware.

Na parte concorrente também se incluem módulos de alarme que identificam a posição da descontinuidade, módulos de memória para o armazenamento destas descontinuidades e um *driver* para a conversão analógica/digital. Também estão incluídos os periféricos de propósito geral (*e.g.*, LCD, UART, botões), os quais são criados pela ferramenta de desenvolvimento (EDK) que cria o *MicroBlaze*.

## 3.1.3 Implementações hardware e software

### 3.1.3.1 Implementação hardware

Para explicar a maneira como os periféricos foram criados, depurados e o seu comportamento simulado, vai-se tomar como exemplo o periférico **Detector**, uma vez que é o periférico mais complexo e importante que se tem. Esclarece-se que este periférico será descrito em detalhe na seção de resultados.

Antes de continuar com a descrição da implementação em hardware é preciso assinalar que a aritmética usada nos periféricos para a detecção das descontinuidades foi aritmética de ponto flutuante em precisão simples segundo o padrão IEEE-754, em que os números reais são representados em palavras de 32 bit (1 bit para o signo, 8 bit para o expoente e 23 para a mantissa) [29]. A razão para esta escolha tem a ver com o caráter genérico do periférico, ou seja, este poderia ser usado para qualquer faixa de valores de entrada, em qualquer outra aplicação que demande detecção de mudança em um sinal. Com isto também se está fazendo uso de resultados de trabalhos prévios [30]. No trabalho prévio ao qual se faz referência foram desenvolvidos periféricos parame-

trizáveis para soma/subtração, multiplicação, divisão e raiz quadrada em aritmética de ponto flutuante [30]. Com a exceção do periférico de raiz quadrada, estes periféricos desenvolvidos foram usados neste trabalho.

Todos os periféricos criados em hardware foram descritos usando o VHDL como a linguagem de descrição hardware. Do periférico **Detector**, por enquanto, pode-se dizer que está composto de sub-periféricos que descrevem as equações da secção 2.4. Alguns destes periféricos contidos na implementação hardware foram implementados usando máquinas de estados FSM (*Finite State Machine*) como controladores do fluxo de dados. A ferramenta de desenvolvimento escolhida foi o ISE 10.1 da Xilinx.

Uma vez que o periférico foi descrito, foram feitas simulações nas quais se consideravam as possíveis entradas e se verificavam suas respostas, determinando-se assim a ocorrência de uma mudança na descrição hardware do periférico. Os tipos de simulações realizadas foram a simulação comportamental e a simulação *Post Place & Route*, as que respectivamente permitem verificar se a estrutura do algoritmo está bem feita, e para verificar o efeito do roteamento no desempenho do periférico. A ferramenta de simulação usada foi o software ModelSim XE da Mentor Graphics.

Existe outro tipo de simulação conhecida com o nome de *VHDL para simulação*. Neste tipo de simulação são usadas opções do VHDL que só funcionam para simulação, porem não são sintetizáveis. A vantagem de este tipo de simulação é a possibilidade de simular uma grande quantidade de dados de entrada de uma maneira simples. Este tipo de simulação só foi feita para o periférico **Detector**, uma vez que é o componente principal e precisava-se conferir o funcionamento do algoritmo de detecção de mudanças em um caso real. Os dados usados nesta simulação foram coletados em testes prévios de soldagem com descontinuidades, mas estes dados não podem ser enviados sem antes serem convertidos para uma representação de ponto flutuante em 32 bit. Estes dados são então enviados para o periférico em teste desde um arquivo de texto. Para quando a simulação termina o resultado é guardado em outro arquivo de texto (para o caso foram dois arquivos, um para o dado do valor estimado e outro para o sinal de alarme), estes arquivos de saída tem seus dados em representação de 32 bit. Para a visualização e análises dos resultados da simulação é feita a conversão para número real. Este método de avaliação, para dados reais de um processo de soldagem é apresentado em [31]

De esta maneira os periféricos ficam prontos para ser baixados na FPGA e ser testados na placa de desenvolvimento.

### 3.1.3.2 Implementação software

A implementação software faz referência ao programa desenvolvido em C para rodar no processador *MicroBlaze* e que tem conta de todo o processo de detecção de descontinuidades. A ferramenta de desenvolvimento usada na implementação software foi o XPS (Xilinx Platform Studio) 10.1 da Xilinx. Em este contexto foram desenvolvidos programas para testar na placa de desenvolvimento periféricos individuais, tanto os periféricos de propósito geral como os periféricos personalizados, e programas para testar o sistemas completo de detecção de descontinuidades.

Com os programas de teste para os periféricos individuais pretende-se verificar o seu comportamento além de definir a forma de seu gerenciamento, ou seja, se for o caso, definir como os dados escolhidos pelo usuário serão enviado, ou como o resultado será apresentado para o usuário. Para a verificação faz-se uso dos módulos LCD e a porta serial da placa de desenvolvimento.

Com os programas para testar o sistema todo, realiza-se o mesmo procediemnto que na verificação individual de periféricos, mas desta vez existem mais periféricos e alguns destes comunicam entre si e a parte software estará relacionado com esta comunicação.

Como foi dito antes, estes programas sequências gerenciam o sistema de detecção das descontinuidades, para isto é preciso executar algumas tarefas e são descritas a seguir:

- Configurar e inicializar os periféricos que vão ser utilizados.
- Configurar e atender o sistema de interrupções.
- Inicializar a interface de usuário (apresentar o sistema ao usuário).
- Apresentar as opções dos parâmetros da bancada de soldagem.
- Enviar a escolha do usuário para os periféricos que correspondam.
- Controlar a bancada de soldagem e a ignição da fonte de soldagem.
- Enviar o sinal que inicia o processo de detecção de descontinuidades.
- Apresentar ao usuário o relatório da detecção de falhas no LCD da placa.
- Se o usuário exigir, será enviado o relatório da detecção das descontinuidades pela porta serial para um PC e logo será armazenado.

### 3.1.4 Configuração do FPGA

O FPGA escolhido para a implementação do sistema de detecção de descontinuidades, no qual incluem-se os algoritmos de detecção de mudança, foi o Spartan-3E 1600 da Xilinx [32], na

tabela 3.1 apresenta-se alguns dos recursos deste FPGA.

**Tabela 3.1.** Alguns recursos da FPGA Spartan-3E 1600 (XC3S1600E).

| Portas | Células lógicas | CLBs | Slices | Multiplicadores |
|--------|-----------------|------|--------|-----------------|
| 1600k  | 33192           | 3688 | 14752  | 36              |

O FPGA vem incorporado na placa de desenvolvimento *Xilinx MicroBlaze Development Kit Spartan-3E 1600E Edition* e é seu principal componente hardware [33]. Esta placa conta com uma variedade de componentes prontos para trabalhar em conjunto com o FPGA, estes são alguns dos componentes da placa:

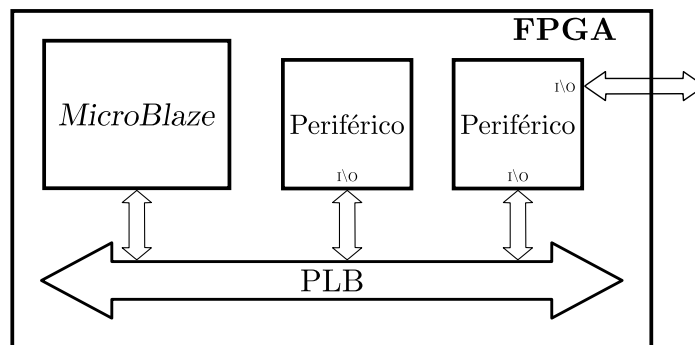
- LCD 2x16
- Porta PS/2
- Porta VGA
- Porta Ethernet 10/100
- Duas portas RS-232
- Oscilador de 50 MHz e 66 MHz
- Três conectores de expansão de 6 pinos
- *Encoder* rotatório com pulsador
- 8 LEDs
- 4 chaves
- 4 pulsadores

Esse tipo de placa de desenvolvimento permite uma prototipagem rápida, os módulos dos que dispõem estas placas está devidamente mapeado aos pinos do FPGA, para usá-los é preciso configurá-los desde a ferramenta de desenvolvimento antes de ser baixados no FPGA.

Este FPGA permite a criação de um processador *soft*, que como anteriormente foi dito, o *MicroBlaze*. A criação do *MicroBlaze* é feita usando o software XPS, durante o processo são definidas a quantidade de memória, a frequência do relógio e alguns dos periféricos da placa que o programa sequencial no *MicroBlaze* utilizará, estes periféricos são os periféricos de propósito geral.

Os periféricos personalizados, os que foram depurados em VHDL, são instanciados no FPGA usando a mesma ferramenta para a criação do *MicroBlaze*. Durante a instanciação são definidos entre outros, o barramento que será usado para a comunicação com o processador e se suas entradas e saídas são internas, ou seja, se estão ligadas ao processador através do barramento escolhido ou se vão fora da FPGA para se comunicar diretamente com algum dos componentes da placa.

A figura 3.2 apresenta um esquema geral da maneira como o processador *MicroBlaze* e os periféricos embarcados na FPGA estão ligados.



**Figura 3.2.** Ligação entre o processador *MicroBlaze* e periféricos personalizados criados em VHDL através do barramento PLB.

Uma vez configurado o FPGA, é possível fazer os testes de verificação e validação.

### 3.1.5 Teste e verificação

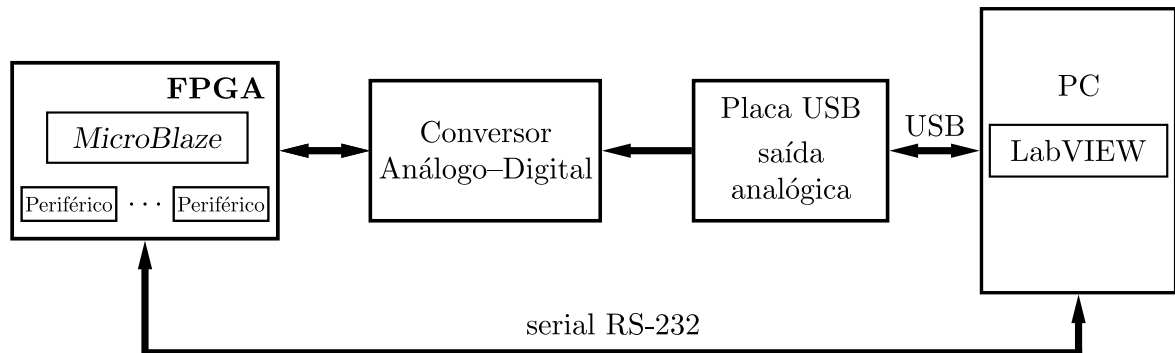
Foram feitos testes com e sem o gerenciamento do *MicroBlaze*. Os testes sem o *MicroBlaze*, foram testes simples de comportamento de algum periférico, nos quais foi suficiente testar o periférico com os estímulos que alguns dos componentes da placa fornecem (*e.g.*, botões, chaves) e a sua resposta foi visualizada nos LEDs ou nos pinos de propósito geral da placa configurados como saídas e conectados a um osciloscópio.

Nos testes que tiveram o gerenciamento do *MicroBlaze*, foram testados e depurados tanto o software no processador quanto a descrição em hardware dos periféricos. Dependendo da complexidade do teste, estes se fizeram unicamente com os recursos da placa, ou seja, usando os botões, os LEDs, a LCD ou as chaves, ou podiam-se fazer usando a porta serial e conectar a placa a um computador. Com esta ultima opção pode-se utilizar ferramentas software como o Hyper Terminal ou LabVIEW. Desta maneira foi possível receber e enviar uma grande e variada quantidade de dados durante a verificação dos componentes desenvolvidos. Assim foram feitos testes de todo o sistema antes de levá-lo para um teste real.

Nos testes em que todo o sistema de detecção foi verificado, simula-se uma experiência de



soldagem real com presença de descontinuidades. Aqui, a saída do sensor é simulada usando o LabVIEW e uma placa capaz de gerar sinais analógicos gerenciados pelo mesmo LabVIEW. A figura 3.3 apresenta o esquema do montagem para testar o sistema de detecção.



**Figura 3.3.** Esquema do montagem para verificação e teste do sistema de detecção.

Antes de descrever como opera o esquema apresentado na figura 3.3, faz-se uma descrição da placa USB e do conversor Análogo/Digital.

**Placa USB:** Esta placa é a placa de aquisição de dados USB-6009 da *National Instruments*, esta placa tem oito canais de entrada analógicos, dois canais de saída analógicos, doze canais entrada/saída digital e um contador de 32 bit, e uma conexão USB [34]. Os conversores Digital/Análogo são de 12 bit e geram valores de tensão de 0-5 V.

**Conversor Análogo/Digital:** Este modulo de conversão, que toma o sinal do sensor infravermelho, é uma placa da Digilent que contém o conversor analógico para digital ADC-S7476MSPS de 12 bits e saída serial, e o filtro *antialiasing* LMV712MM de dois pólos [35]. Esta placa é conectada a placa de desenvolvimento através de uma de suas portas de expansão. O *driver* para este conversor é descrito na secção de resultados.

Voltando para os testes feitos segundo o esquema da figura 3.3. Os dados usados nestes testes derivam de experiências prévias e são carregados pelo LabVIEW e enviados um de cada vez à placa USB que gera o sinal analógico correspondente para cada dado. A saída analógica da placa USB é ligada à entrada do conversor Analógico/Digital que por sua vez esta ligado à placa de desenvolvimento. Existe um protocolo de comunicação através da porta serial para sincronizar o envio dos dados desde o PC para serem lidos na placa de desenvolvimento. O software no *MicroBlaze* inicializa os periféricos e solicita o dado para o PC. Este coloca o dado na placa USB e informa para o *MicroBlaze* que é permitido adquirir o dado do sensor para se fazer a estimativa dele, uma vez que a estimativa foi calculada e foi determinado se houver alguma descontinuidade, o seguinte dado é solicitado para o PC, desta maneira até que seja atingida o total de dados da simulação.

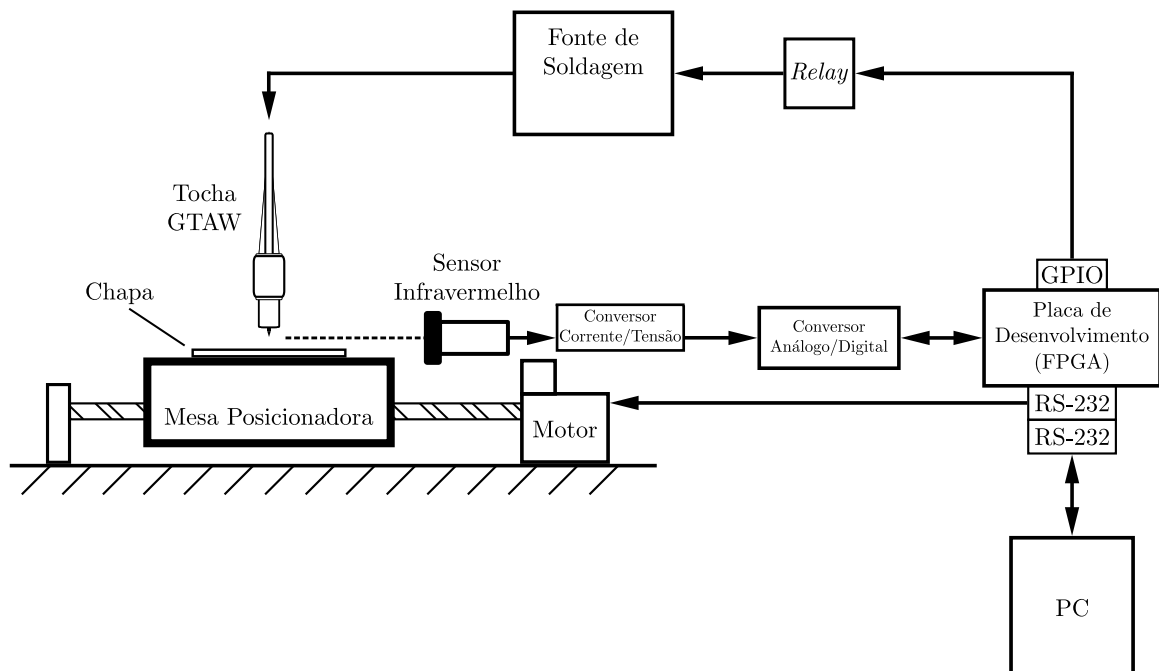
Se a simulação não atinge a resposta desejada, é necessário ocorrer mudanças seja no programa que roda no *MicroBlaze* ou na descrição hardware dos periféricos como sugere o diagrama de blocos da figura 3.1 e executar novamente o processo aqui descrito até alcançar um resultado para ser testado no sistema de soldagem.

A idéia deste procedimento e a montagem da figura 3.3 é ter um sistema que permita verificar e testar todo o sistema de detecção inteiro sem ter que ir até a bancada de soldagem a cada modificação feita no sistema de detecção, permitindo ganhar tempo no desenvolvimento do sistema e poupar material das provas (*i.e.*, corpos de prova e gás de proteção principalmente).

Desta forma foram feitos os testes para levar o sistema de detecção no ponto exato. A descrição de cada parte do sistema de detecção das descontinuidades é apresentado na secção de resultados.

### 3.2 SISTEMA DE SOLDAGEM.

Uma vez que o sistema de detecção esteja pronto, segundo os resultados das simulações, deve ser testado em provas reais que são feitas no *sistema de soldagem*, mas antes de começar a descrever os teste é favorável conhecer o sistema de soldagem. Como é apresentado nas figuras 3.4 e 3.5 o sistema de soldagem esta composto por:



**Figura 3.4.** Diagrama esquemático do sistema de soldagem.

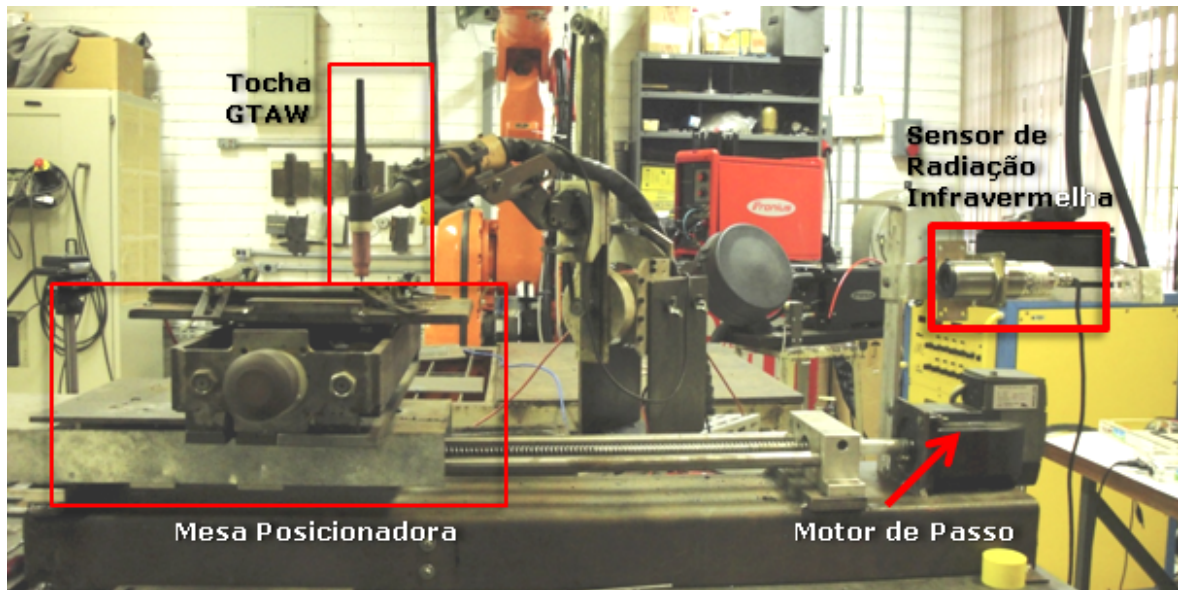
- Mesa posicionadora.
- Tocha GTAW (Fonte de soldagem)
- Sistema de sensoriamento.
  - Voltímetro.
  - Pinça amperimétrica.
  - Sensor de infravermelho.
- Computador

A **Mesa Posicionadora** é uma plataforma que se pode deslocar linearmente e onde é colocada e assegurada a peça que será soldada. O deslocamento da mesa é transmitido por um motor de passo através de um parafuso sem fim. A velocidade, o tempo de deslocamento e a direção de deslocamento da mesa podem ser ajustados, o que permite ter controle do início e fim dos cordões de solda, estes parâmetros são enviados para o controlador da mesa desde o FPGA via RS-232.

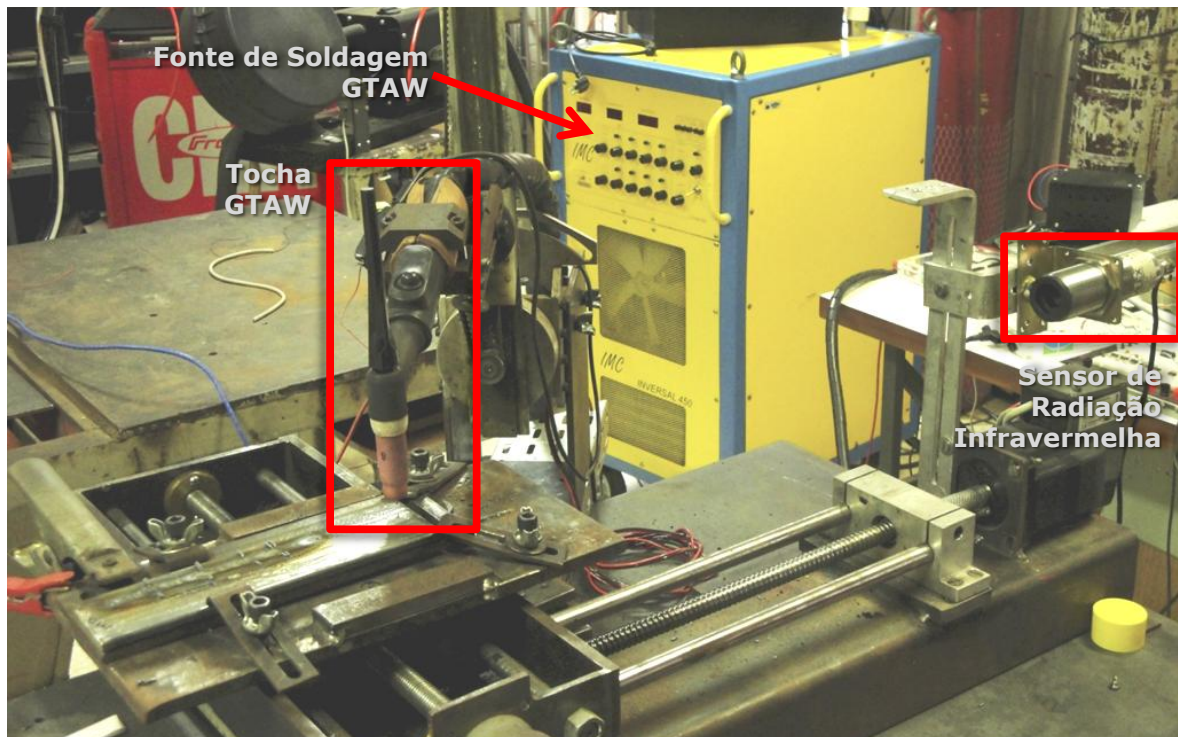
A **Tocha GTAW** é colocada por acima da mesa posicionadora e a peça a ser soldada, e permanece estacionaria durante todo o processo de soldagem, o eletrodo é ajustado para *stand-off* de 5 mm ou 3 mm durante os testes. A técnica usada para abrir o arco é mediante o encostamento do eletrodo na peça, e a técnica para mantê-lo ligado é a conhecida como de “dois toques” que consiste em deixar ligado o gatilho da tocha, ou seu equivalente na fonte de soldagem, durante o tempo que dura a solda. A ativação do gatilho ou ignitor da fonte de soldagem é feita desde o FPGA usando como acoplamento um *relay*.

O **Sistema de Sensoriamento** tem base em sensores para tensão e para corrente. O voltímetro é conectado entre o eletrodo e a peça que a ser soldada, medindo assim a tensão do arco. Este sensor tem um sistema de isolamento antes de ir para a placa de aquisição de dados. O sensor usado para medir a corrente é uma pinça amperimétrica cuja saída vai conectada diretamente para a placa de aquisição de dados. Para cumprir o propósito de este trabalho, é adicionado ao sistema de soldagem o sensor óptico de infravermelho. O sensor infravermelho usado foi o TL-GA-18-1 da Calex [36], a tabela 3.2 apresenta algumas especificações do sensor. O sensor de infravermelho é um pirômetro óptico que tem seu foco a 60 cm, e conta com um laser que aponta na direção do foco, para ajudar na localização do sensor em linha reta ao arco durante a solda. O sensor foi localizado de maneira perpendicular à tocha com o propósito de coletar mais da radiação produzida pelo arco do que pela poça de fusão. O valor do sinal de saída deste sensor vai de 4 mA até 20 mA e é proporcional a radiação capturada.

Este sensor é utilizado cotidianamente para medir temperatura, mas neste trabalho não é de interesse medir esta grandeza no arco, o interesse esta na variação que possa apresentar o



(a) Vista frontal do sistema de soldagem



(b) Vista do sistema de soldagem que permite ver a fonte de soldagem.

**Figura 3.5.** Fotografias que apresentam o sistema de soldagem e como a suas partes estão dispostas.

sinal medido (até aqui, corrente), por isso a medida bruta do sensor é utilizada sem que seja transformada por uma função de transferência.

O sinal de saída do sensor infravermelho (corrente) é ligada a um conversor corrente/tensão, configurado para uma saída de tensão entre 0 e 5 V [37]. Por sua vez, este sinal de tensão é

**Tabela 3.2.** Algumas especificações técnicas do sensor TL-GA-18-1

| Parâmetro             | Valor                  |
|-----------------------|------------------------|
| Faixa de medição      | 350-1800°C             |
| Faixa de espectral    | 1,45-1,8 $\mu\text{m}$ |
| Tensão de alimentação | 24 VDC                 |
| Sinal de saída        | 4-20 mA                |

conectado ao conversor Analógico/Digital que coleta o sinal para o detector de descontinuidades embarcado no FPGA.

O *Computador* do sistema de soldagem conta com uma interfase LabVIEW no qual é possível fazer o gerenciamento de todo o sistema de soldagem. Este esquema é usado quando ocorrem testes off-line de monitoramento, para testar parâmetros de solda, caracterização e para propósito geral segundo as possibilidades da bancada, mas para quando são executados os testes com o FPGA e a placa de desenvolvimento, o computador tem o papel de acompanhar o processo e uma vez terminado o teste de soldagem recebe desde a placa de desenvolvimento aos dados das descontinuidades encontradas que são enviados por o *MicroBlaze* pela porta serial.

### 3.2.1 Uso do sistema de soldagem.

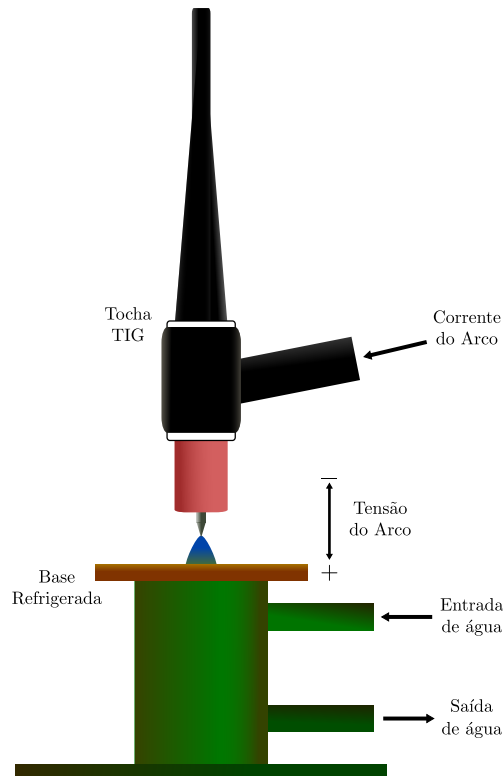
Houveram dois usos principais no sistema de soldagem: no primeiro deles o sistema foi usado para a etapa de caracterização do processo de soldagem, no segundo foi usado para a etapa de provas e verificação.

#### 3.2.1.1 Etapa de caracterização

Esta etapa inicia com a caracterização do arco GTAW. O propósito de esta caracterização é encontrar uma região estável de operação da fonte de soldagem e fazer os testes. Na figura 3.6 apresenta-se a bancada usada para o levantamento da curva. Como a figura 3.6 mostra, a tocha GTAW esta exatamente acima da base refrigerada por água, o que permite medir a tensão e a corrente do arco sem preocupar-se com o tempo que este procedimento consuma uma vez que não formará poça de fusão.

O procedimento para a obtenção da curva característica compreende o seguintes passos:

1. Configurar a bancada para a caracterização (conectar os sensores, a fonte de soldagem e o água).



**Figura 3.6.** Bancada para obter a curva característica do arco GTAW.

2. Fixar a distancia eletrodo base refrigerada (*stand-off*).
3. Definir a faixa de corrente (corrente mínima e corrente máxima) na que vai ser feito o procedimento de caracterização, e como vai ser o incremento deles.
4. Fixar a fonte de soldagem para iniciar desde o valor mínimo de corrente.
5. Iniciar o procedimento abrindo o arco desde o valor de corrente menor.
6. Uma vez que o arco foi aberto é necessário esperar um tempo para que o arco fique estável.
7. Coletar os dados da corrente e tensão para a corrente fixada e obter com eles o primeiro ponto da curva.
8. Aumentar o valor de corrente para o próximo e fazer os mesmos passos desde 6, e de esta maneira continuar até o último valor de corrente predefinido.
9. Finalmente obtém-se a curva característica do arco com um número de pontos igual ao número de valores de corrente fixados durante o procedimento.

O procedimento anteriormente numerado foi feito para valores de *stand-off* de 3 e 5 mm. Durante a caracterização foi utilizado o sensor de infravermelho para acompanhar as medidas de corrente e tensão. Na seção de resultados as curvas obtidas serão apresentadas.

Uma vez definida a região de estabilidade da fonte pela curva característica, procede-se à obtenção do padrão do cordão de solda a partir de cordões sem nenhum tipo aparente de descontinuidade, usando para isto a bancada de soldagem das figuras 3.4 e 3.5, mas gerenciada pelo computador. Estes cordões mantiveram constantes o *stand-off* (3 ou 5 mm), a velocidade de deslocamento (2,5 mm/s) e a vazão de gás (15 L/min), mas usando valores diferentes de corrente, sendo estes os valores próximos aos extremos e ao valor médio da região estável segundo a curva característica do arco, assim 80 A, 140 A e 180 A.

Deve-se lembrar que o sinal de interesse para o padrão de solda é o sinal do sensor infravermelho dado que é este irá como a entrada para o sistema de detecção de descontinuidades embarcado no FPGA. Com o resultado das análises dos dados da solda padrão o valor de  $R$  para o sistema de detecção de descontinuidades é obtido. Segundo a técnica de detecção de mudança que será usada para detectar descontinuidades na soldagem, este padrão seria a hipótese nula  $H_0$  (apresentada na seção ??), que diz que não houve descontinuidades no sinal.

### 3.2.1.2 Etapa de provas e verificação

A primeira parte desta etapa é dirigida à coleta de dados de corrente, tensão e radiação infravermelha durante a produção de cordões de solda, onde intencionalmente foram introduzidas uma variedade de descontinuidades. Durante este procedimento, a bancada de soldagem é gerenciada desde o computador. A idéia com estes testes é usar os dados para ajustar os parâmetros do sistema de detecção das descontinuidades e verificar de maneira *off-line* a resposta do sistema, ou seja, verificar se é possível detectar as descontinuidades. Para esta verificação usa-se o software *MatLab*, que permite ler os dados da radiação infravermelha de todas as provas, atribuir valores para  $Q$  e  $h$  e rodar uma versão software do algoritmo de detecção de mudança e de esta maneira determinar se o ajuste é adequado para a detecção das descontinuidades e ajustar à descrição hardware.

A parte final desta etapa no uso do sistema de soldagem, é a parte de validação de todo o sistema de detecção de descontinuidades. As provas de soldagem feitas nesta etapa estão gerenciadas desde a placa de desenvolvimento onde está embarcado o sistema de detecção. O computador acompanha o processo e no final através da porta serial recebe a informação de quantas descontinuidades aconteceram e o lugar onde elas aconteceram para armazená-las em um arquivo de texto. As provas de soldagem são realizadas da mesma maneira como foram descritas no parágrafo anterior, algumas descontinuidades são adicionadas intencionalmente, mas esta vez, elas são detectadas enquanto o processo de soldagem acontece, ou seja, são detectadas de maneira *on-line*.

Os teste para a validação do sistema de detecção, nos que intencionalmente foram introduzidas descontinuidades, são apresentados na seguinte seção.

### 3.3 TESTES PARA VALIDAÇÃO.

Para a avaliação tanto do *sistema de detecção* como do *sistema de soldagem* foram feitos uma serie de testes de soldagem (cordões de solda) com presença intencional de descontinuidades. Estes testes de soldagem foram feitos com os parâmetros apresentados na tabela 3.3. As

**Tabela 3.3.** Parâmetros de soldagem para os testes de validação.

| Parâmetro        | Valor         |
|------------------|---------------|
| Corrente         | 140 A         |
| Velocidade       | 2,5 mm/s      |
| Tempo            | 48 s          |
| <i>Stand-off</i> | 3 e 5 mm      |
| Eletrodo         | EWTh-2; 1,6mm |
| Vazão de gás     | 12 L/min      |

descontinuidades nos testes de solda foram alocadas ou fixadas no percurso do cordão de solda em diferentes lugares respeito do inicio da solda, sendo de três tipos diferentes: arame, furos e areia. Considera-se que estes tipos de descontinuidades representam um grupo de descontinuidades que se podem encontrar em um processo soldagem automatizado em campo ou industrial.

**Arame:** O arame representa uma protuberância na peça que é soldada. O arame usado é uma mistura de metais com base em aço e um diâmetro de 1,2 mm. Por cada descontinuidade de um até três destes arames foram alocados, ou seja, havia um, dois ou três arames por cada descontinuidade.

**Furos:** O furo representa o contrário à protuberância na peça soldada. Estes furos foram feitos nas chapas, no percurso do cordão de solda. Cada furo tem um diâmetro entre 6 e 10 mm e uma profundidade aproximada de 4 mm.

**Areia:** A areia é usada como um exemplo de sujeira na peça que é soldada. No cordão de solda foram alocados trechos de areia com 5 até 15 mm aproximadamente de comprimento. Esclarece-se que não são montículos de areai, esta foi espalhada pela peça.

O sistema de detecção de descontinuidades foi avaliado para cada tipo de descontinuidade. Os resultados de estes testes se apresentam na seção de resultados.



## 4 RESULTADOS

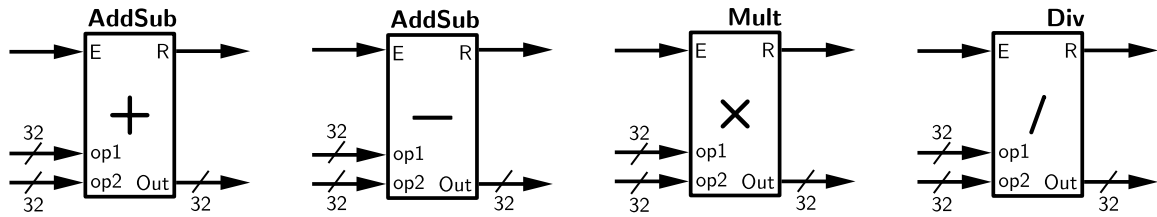
Nesta seção apresentam-se os resultados obtidos no desenvolvimento deste trabalho em duas partes . A primeira *Resultados Hardware*, descreve os periféricos desenvolvidos em hardware para o sistema de detecção de descontinuidades, a segunda *Resultados de soldagem* encontram-se os resultados da validação do sistema em campo, ou seja, os resultados do sistema na detecção das descontinuidades em um processo de soldagem GTAW.

### 4.1 RESULTADOS HARDWARE.

Segundo os objetivos propostos deste trabalho implica-se descrever em hardware o algoritmo para a detecção de mudanças e os módulos adicionais que são complemento do sistema de detecção das descontinuidades. Desta maneira foram desenvolvidos uma série de periféricos que se descrevem baixo os seguintes campos:

- Conversão análoga/digital.
- Detector de mudanças.
- Conversão e detecção.
- Gerenciador do tempo.
- Memória de descontinuidades.
- Gerenciamento do *MicroBlaze*.

Alguns dos periféricos mencionados anteriormente fazem uso de periféricos especiais para soma/subtração, multiplicação e divisão, que não serão descritos neste trabalho, pois são resultado de um trabalho prévio [30]. A figura 4.1 apresenta o esquema de cada um deles.



**Figura 4.1.** Esquema que representa os periféricos de soma/subtração multiplicação e divisão usados em alguns dos periféricos desenvolvidos. Relembrando que o periférico de soma e subtração é o mesmo, o tipo de operação é definido por um sinal de entrada, que para simplificar não aparece desenhada nos esquemas.

Estes periféricos funcionam da seguinte maneira: Os dados que serão operados entram pelas portas *op1* e *op2*, para iniciar o cálculo a entrada *E* deve estar em 1. Ao término do cálculo o resultado é colocado na porta *Out* e a porta *R* é 1 indicando que o cálculo está pronto e pode ser lido.

#### 4.1.1 Conversão análoga/digital.

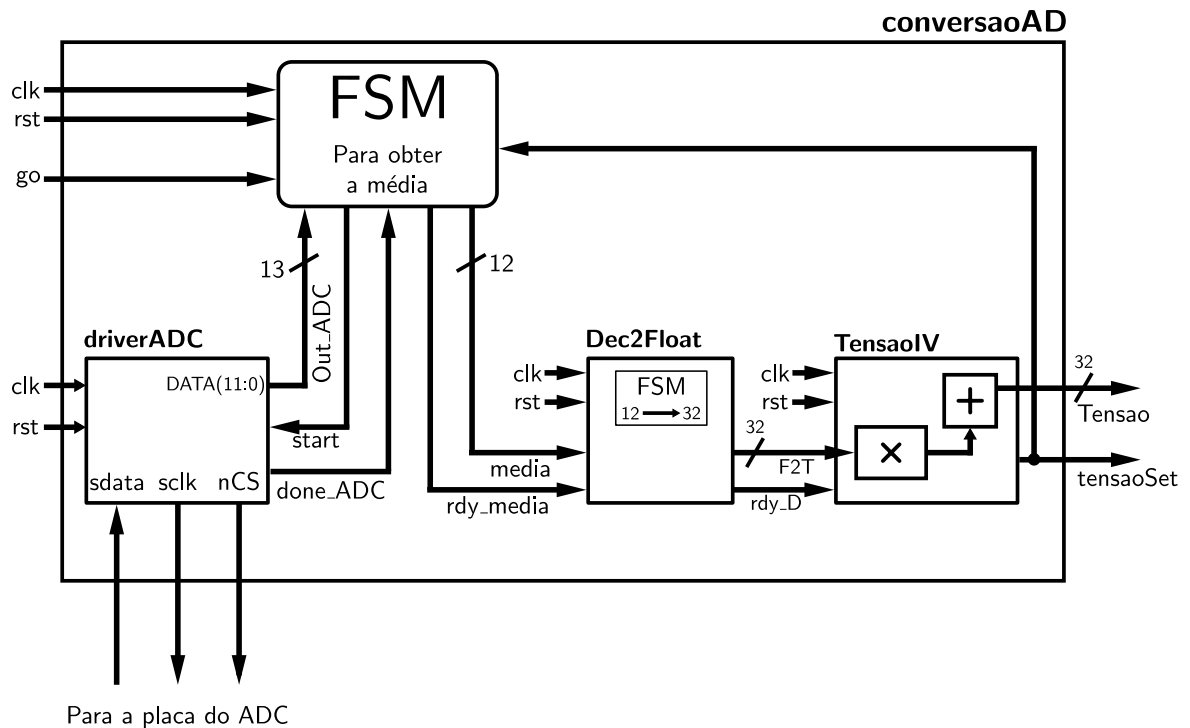
Como foi dito anteriormente, além da placa de desenvolvimento que contém o FPGA Spartan-3E 1600, foi usada uma placa da Digilent para a conversão análoga/digital [35] (descrita na seção 3.1.5). Esta placa de conversão conecta-se à placa de desenvolvimento através de um dos conectores entrada/saída de propósito geral que ela possui. O *driver* para esta placa de conversão, aqui chamado de **driverADC**, foi fornecido pelo fabricante, mas para os propósitos deste trabalho sofreu algumas modificações. O **driverADC** recebe o dado da conversão análoga/digital serialmente e entrega-o em um arranjo de 12 bit. Mas este resultado em 12 bit tem que ser transformado para 32 bit e enviado à etapa de detecção de mudança.

Para logar criou-se o periférico **conversaoAD**, que além do **driverADC**, possui os seguintes sub-periféricos: **Dec2Float** e **TensaoIV**.

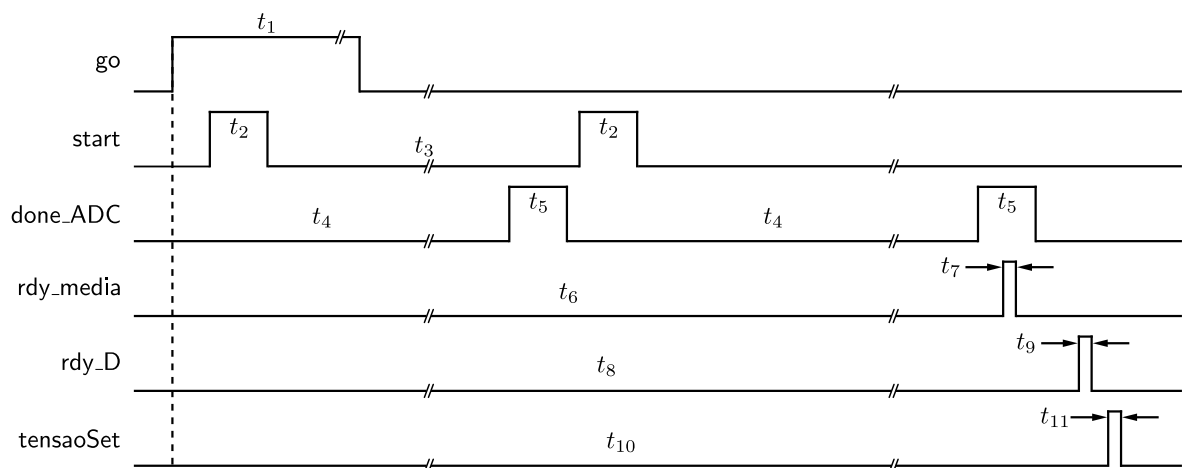
**Dec2Float:** Este periférico tem uma máquina de estados FSM que recebe um dado inteiro de 12 bit e o transforma para seu equivalente em ponto flutuante em 32 bit.

**TensaoIV:** Este periférico recebe o valor da conversão análoga/digital em 32 bit e entrega um valor correspondente à tensão medida em 32 bit.

A maneira como estes periféricos estão ligados entre eles para conseguir o valor de tensão medido é apresentado na figura 4.2



**Figura 4.2.** Diagrama da arquitetura do periférico **conversaoAD**, que faz a conversão análoga/digital do sinal do sensor infravermelho e entrega o resultado em 32 bit.



**Figura 4.3.** Diagrama de resposta no tempo para o periférico **conversaoAD**, o tempo consumido desde a ativação do periférico até que o resultado está disponível é  $t_{10} = 3311$  ns.

Como a figura 4.2 apresenta, existe uma máquina de estados (FSM) que gerencia o periférico **conversaoAD**. O processo inicia quando o sinal de entrada **go** é ativado. A FSM lê dois dados diferentes de conversão de **driverADC** soma-os em um arranjo de 13 bit, desloca-o para a direita, toma os 12 bit menos significativos de forma a obter uma média de aqueles dois dados. O resultado da média é enviado pelo sinal **media** para o periférico **Dec2Float**, que inicia a conversão

para 32 bit quando o sinal `rdy_media` é ativado pela FSM. O dado da conversão em 32 bit é alocado no sinal F2T e vai para o periférico **TensaoIV**. Uma vez que se tem o valor de tensão da conversão analógica/digital no sinal **Tensao** o sinal `tensaoSet` é ativado, ficando a FSM pronta para uma próxima conversão ao mesmo tempo que o periférico **conversaoAD** informa para a seguinte etapa que o valor da conversão já esta pronto e pode ser lido.

Como complemento à descrição do periférico **conversaoAD**, são apresentadas na figura 4.3 as respostas no tempo dos sinais mais relevantes do periférico, onde pode-se conhecer quanto tempo o periférico consome em uma operação. Na tabela 4.1 estão descritos cada um dos tempos apresentados na figura 4.3.

**Tabela 4.1.** Descrição dos tempos consumidos pelos sinais mais relevantes do periférico **conversaoAD**.

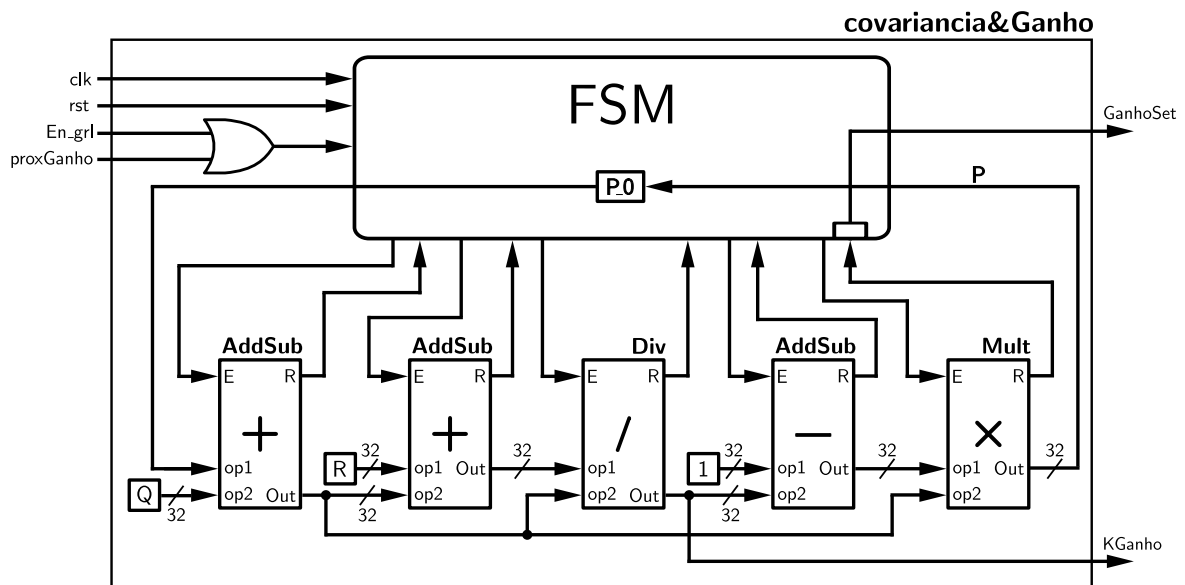
| Intervalos de tempo | Duração (ns) | Descrição  |
|---------------------|--------------|--|
| $t_1$               | 220          | Duração do sinal de início dos processos do periférico enviado desde o <i>MicroBlaze</i> .                                   |
| $t_2$               | 80           | Tempo para iniciar a primeira conversão analógica/digital usada para obter a média.  |
| $t_3$               | 1500         | Tempo consumido até iniciar a próxima (e última) conversão analógica/digital para obter a média.                             |
| $t_4$               | 1537         | Tempo consumido para obter o resultado de uma conversão analógica/digital.   |
| $t_5$               | 80           | Duração ativa do sinal que indica que a conversão analógica/digital terminou.  |
| $t_6$               | 3170         | Tempo consumido para obter o valor da média da medição.  |
| $t_7$               | 20           | Duração ativa do sinal que indica o valor da média já se tem e o periférico <b>Dec2Float</b> pode iniciar.                   |
| $t_8$               | 3272         | Tempo que levou obter o valor em 32 bit do dado medido desde que o periférico foi ativado.                                   |
| $t_9$               | 20           | Duração ativa do sinal que indica o valor da conversão esta em pronto em 32 bit e o periférico <b>TensaoIV</b> pode iniciar. |
| $t_{10}$            | 3311         | Tempo total consumido pelo periférico para obter o valor de tensão do dado de entrada.                                       |
| $t_{11}$            | 20           | Duração ativa do sinal indicando que o dado da tensão esta pronto e pode ser usado.  |

## 4.1.2 Detector de mudanças.

A descrição hardware do periférico que contém o algoritmo do detector de mudança, que inclui o filtro Kalman e a regra de parada com o teste CUSUM, esta contida no periférico chamado **Detector** e baseia-se no descrito na seção 2.4. Por sua vez este periférico contém três sub-periféricos para ajudar no seu entendimento e depuração, sendo assim: **covariancia&Ganho**, **estimador&erro** e **CUSUM&limiar**. Cada um destes periféricos são descritos a seguir.

### 4.1.2.1 covariancia&Ganho

O periférico **covariancia&Ganho** que é apresentado na figura 4.4, calcula o valor da variância da próxima iteração segundo a equação (2.12) e o ganho de Kalman segundo a equação (2.14). Como a figura apresenta, o fluxo de dados no periférico é gerenciado por uma máquina de estados FSM, que também contém representado em 32 bit o valor inicial  $P_0$  e as constantes  $Q$ ,  $R$  e  $1$ . Este periférico é um daqueles que fazem uso dos periféricos de soma/subtração, multiplicação e divisão em 32 bit.



**Figura 4.4.** Diagrama da arquitetura do periférico **covariancia&Ganho** que é ativado ou por o sinal de entrada  $En\_grl$  ou por o sinal de entrada  $proxGanho$ , e gera o valor da contante do filtro no sinal de saída  $KGanho$ .

Este periférico é ativado ou por o sinal de entrada  $En\_grl$ , sinal que é enviado uma única vez, ou por o sinal de entrada  $proxGanho$  que vem do periférico **estimador&erro**. As saídas deste periférico são: o sinal  $GanhoSet$  que indica que o periférico terminou a sua parte e o valor do ganho de Kalman esta já disponível no outro sinal de saída de 32 bit  $KGanho$  e pode ser utilizado, estes dois sinais de saída vão ser usadas pelo periférico **estimador&erro** no seus cálculos. O sinal

de saída **GanhoSet**, é ativada pela FSM só depois que o periférico **Mult** termina de fazer seu cálculo ( $R=1$  em **Mult**). Além da descrição realizada até aqui, a FSM atualiza o valor de  $P_0$  por  $P$ , que na próxima iteração será usado como o dado prévio da variância.

Dado que o ganho de Kalman é um parâmetro *off-line*, o periférico **covariancia&Ganho** pode ser ativado quando o software que roda no *MicroBlaze* está na etapa de inicialização, e seu primeiro resultado do ganho de Kalman a ser usado quando tenha-se a primeira medida do sensor infravermelho e o processo de estimação da medição inicie.

Na figura do periférico **covariancia&Ganho** apresentam-se dois blocos de soma e um de subtração, mas o nome dos blocos é o mesmo, **Addsub**. Isto indica que só foi feita uma instanciação deste componente, e usado duas vezes como soma e mais uma vez como subtração.

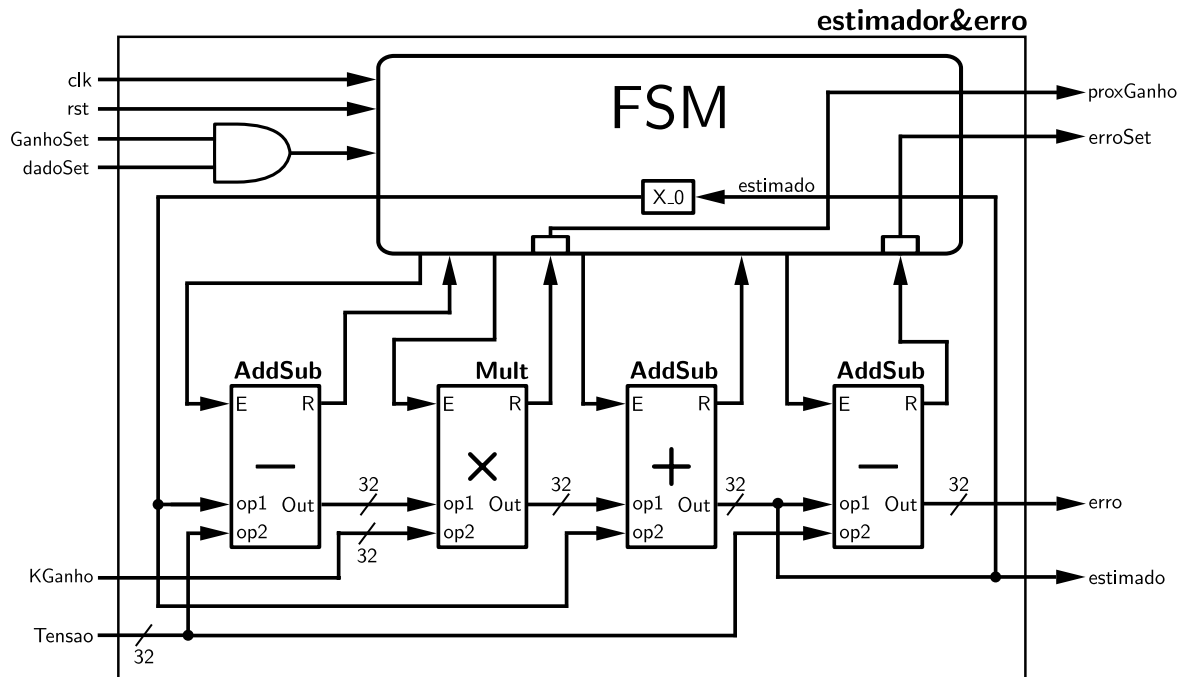
#### 4.1.2.2 **estimador&erro**

O diagrama da arquitetura do periférico **estimador&erro** é apresentado na figura 4.5, este periférico calcula o valor estimado do dado de entrada segundo a equação (2.13) e a diferença entre eles segundo a equação (2.17). Este periférico também conta com uma FSM que gerencia o seu fluxo de dados. O sinal de entrada **Tensao**, vem do periférico **converAD** e tem em 32 bit o resultado da conversão análoga/digital, o outro sinal de entrada **KGanho** vem do periférico descrito anteriormente **covariancia&Ganho**, com estes sinais de entrada e o valor inicial  $X_0$  o periférico faz seus cálculos.

Este periférico é ativado por o sinal de entrada **dadoSet**, que indica o outro sinal de entrada **Tensao** esta pronto e pode se fazer a sua estimação, e o sinal de entrada **GanhoSet** que vem do periférico **covariancia&Ganho**. O periférico tem quatro saídas, a primeira a ser ativada pela FSM é o sinal **proxGanho**, uma vez o periférico **Mult** termina seu cálculo, fazendo que o periférico **covariancia&Ganho** seja ativado de novo enquanto o processo dos cálculos no periférico **estimador&erro** continuam e gerar as outras três saídas.

O sinal de saída **estimado** contém o valor estimado do dado de entrada e irá para o periférico **CUSUM&limiar**, que também receberá os demais sinais de saídas; **erro** e **erroSet**, este último sinal é ativado pela FSM uma vez que o último periférico de subtração termina seu cálculo. A FSM também atualiza o valor de  $X_0$  com o valor de **estimado** para ser usado na próxima iteração.

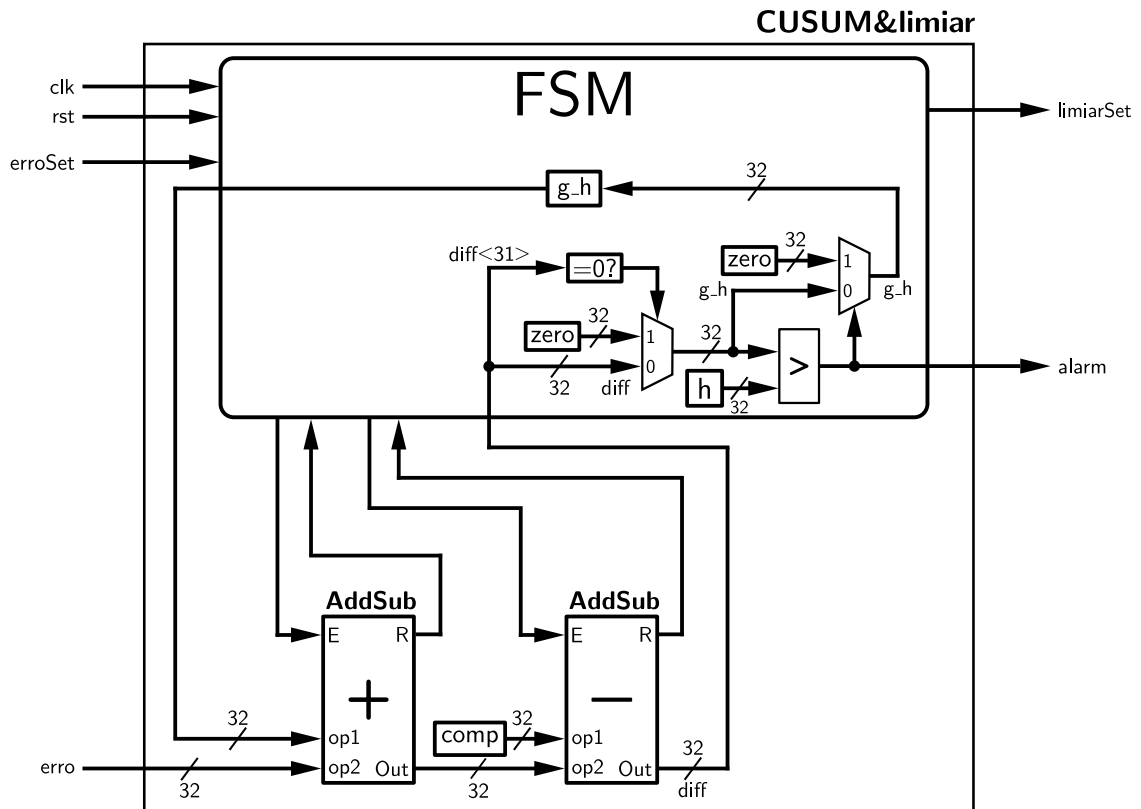
Este periférico também tem instanciado o periférico **Addsub** uma vez e usado dois vezes para subtração e uma vez para soma.



**Figura 4.5.** Diagrama da arquitetura do periférico **estimador&erro** que é ativado por **dadoSet** e **GanhoSet** e gera o valor estimado do filtro estimado

#### 4.1.2.3 CUSUM&limiar

O diagrama da arquitetura do periférico **CUSUM&limiar** é apresentado na figura 4.6, este periférico contém a descrição hardware da regra de parada do detetor de mudança e que foi apresentada na seção ??, ou seja, que é este periférico que ativa o sinal de alarme sempre que a soma cumulativa ultrapassa o limiar predefinido. Como a figura apresenta o fluxo de dados deste periférico também é gerenciado por uma FSM que tem em 32 bit o valor das constantes do limiar  $h$ , o compensador  $comp$  e zero, igualmente tem o valor inicial da soma cumulativa ou teste estatístico  $g\_h$ . Este periférico está alocado em série depois do periférico **estimador&erro**, e inicia seus cálculos tomando o sinal de entrada **erro** só quando o sinal de entrada **erroSet** é ativado. Com o bit mais significativo (o bit de signo) do resultado da subtração ( $diff$ ) determina-se se a soma cumulativa vai para zero (se  $diff < 31 > = 1$  então  $g\_h = zero$ ) ou continua-se acumulando (se  $diff < 31 > = 0$  então  $g\_h = diff$ ). Depois a FSM compara o valor na soma cumulativa  $g\_h$  com o limiar  $h$ , se  $g\_h > h$  o sinal de saída **alarme** é ativado e  $g\_h = zero$ , se não  $g\_h$  não muda, atualizando assim o valor da soma cumulativa para a próxima iteração. Depois disso a FSM indica ativando **limiarSet** que o periférico termino sua parte.



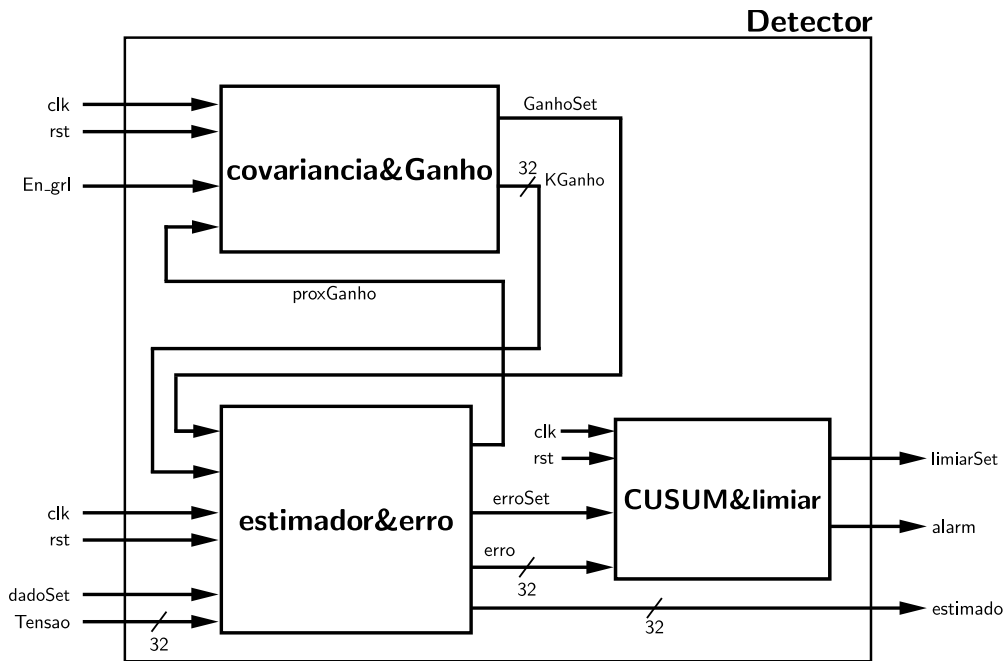
**Figura 4.6.** Diagrama da arquitetura do periférico **CUSUM&limiar** que é ativado por o sinal de entrada **erroSet** para gerar a alarme, se houver, a partir do sinal de entrada **erro**.

Como foi apresentado previamente os três periféricos descritos acima são sub-periféricos de um componente maior, o periférico **Detector**. A maneira como os sub-periféricos estão conectados entre eles é esquematizado na figura 4.7

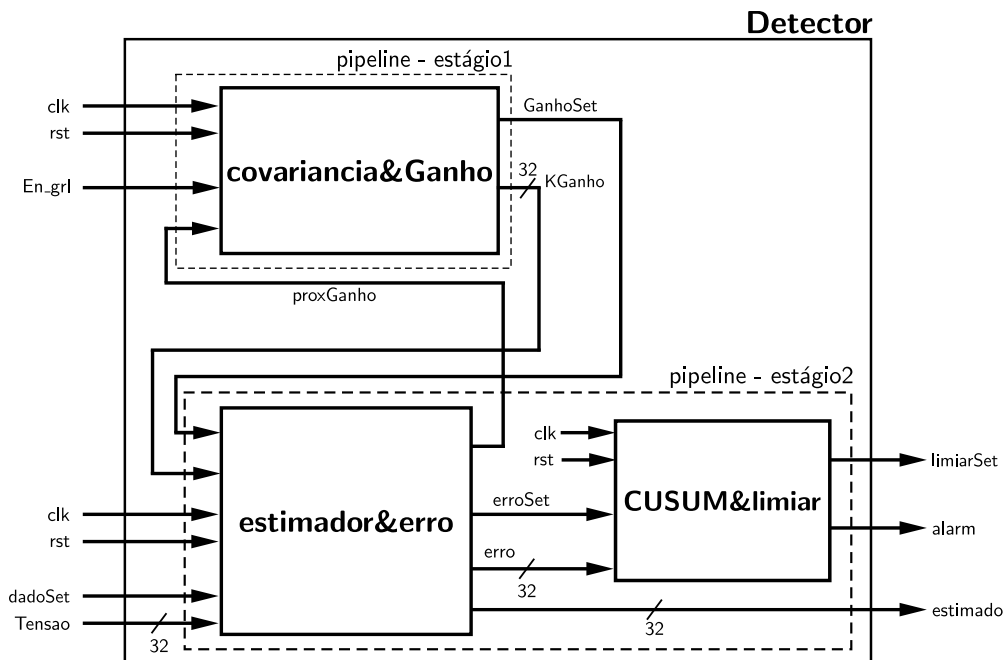
A figura 4.7 mostra como os sub-periféricos são executados e como é o fluxo de dados na execução deles. Quando o sinal de entrada **dadoSet** é ativado, o sinal **KGanho** já tem o dado do cálculo realizado pelo periférico **covariancia&Ganho** para ser usado na presente iteração. No médio da execução do periférico **estimador&erro** o sinal **proxGanho** é ativada iniciando desta maneira o cálculo do ganho de Kalman que será usado na próxima iteração. Uma vez que a execução do periférico **estimador&erro** termina, a execução do periférico **CUSUM&limiar** inicia para determinar se uma descontinuidade foi encontrada ativando o sinal de alarme, quando a execução termina o sinal **limiarSet** e ativada e o periférico **Detector** fica pronto para a execução da seguinte iteração, para a qual já tem pronto o valor do ganho de Kalman.

O fato, como foi descrita a execução do periférico **Detector**, define a arquitetura de uma estrutura *pipeline* de dois estágios como é especificado na figura 4.8.



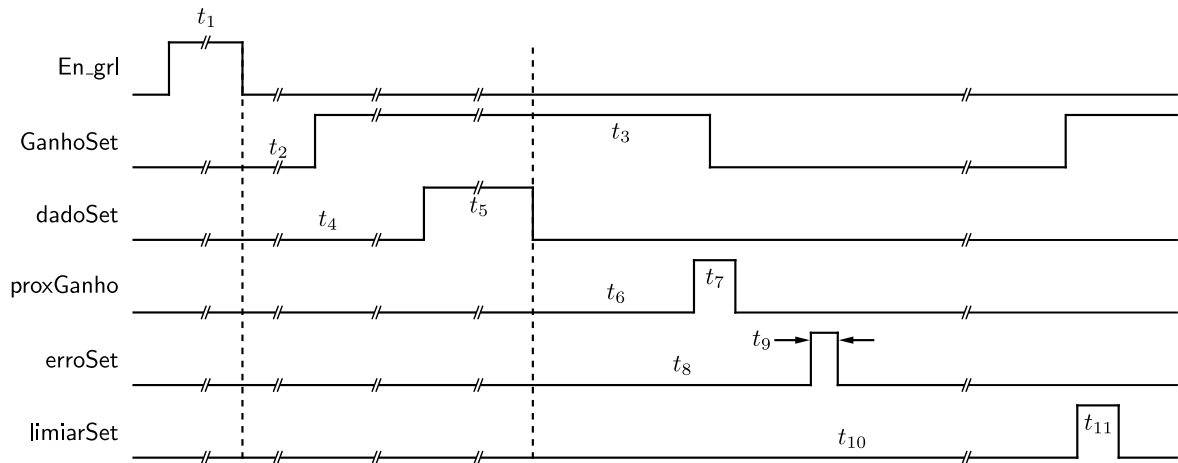


**Figura 4.7.** Diagrama de blocos do periférico **Detector** que apresenta a maneira como os periféricos **covariancia&Ganho**, **estimador&erro** e **CUSUM&limiar** foram interligados.



**Figura 4.8.** Diagrama de blocos do periférico **Detector** visto como uma estrutura *pipeline* de dois estágios.

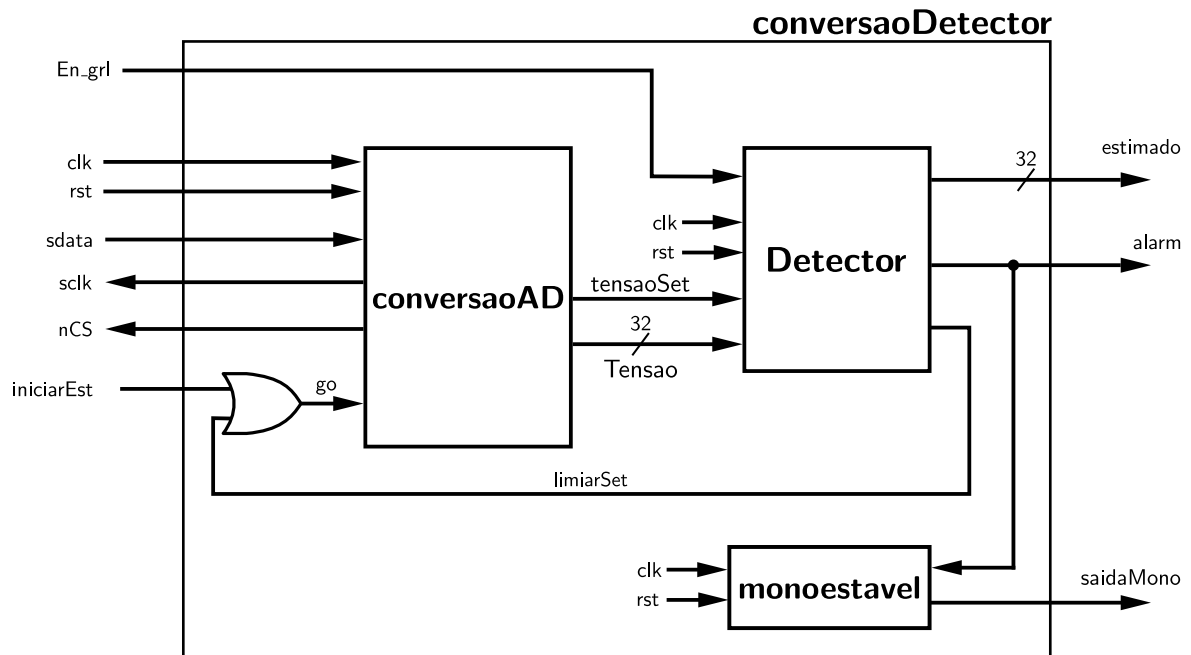
Para o periférico **Detector** também é apresentada as respostas no tempo de alguns de seus sinais mais relevantes como uma maneira de entender melhor o seu funcionamento, isto pode ser visto na figura 4.9. A descrição dos tempos que apresentam-se na figura estão na tabela 4.2.



**Figura 4.9.** Diagrama de resposta no tempo para o periférico **Detector**, o tempo consumido desde a ativação do periférico pelo flanco de decida do sinal **dadoSet** até que o resultado esta disponível é  $t_{10} = 850$  ns.

### 4.1.3 Conversão e detecção

A seguir se apresenta e descreve o primeiro grande periférico desenvolvido para ser gerenciado diretamente pelo *MicroBlaze*. O periférico foi chamado de **conversaoDetetor** contém os periféricos **conversaoAD** e **Detector** descritos anteriormente, e o periférico **monoestavel** que é a implementação de um monoestável reativável. A figura 4.10 apresenta a arquitetura do periférico **conversaoDetetor**.



**Figura 4.10.** Diagrama da arquitetura do periférico **conversaoDetetor**. Contem como sub-periféricos **conversaoAD**, **Detector** e **monoestavel**

**Tabela 4.2.** Descrição dos tempos consumidos pelos sinais mais relevantes do periférico **Detector**.

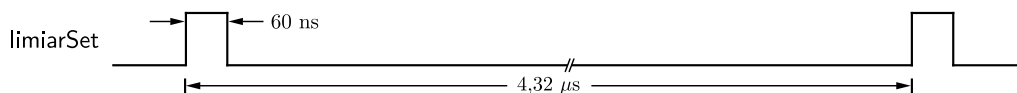
| Intervalos de tempo | Duração (ns) | Descrição  |
|---------------------|--------------|--|
| $t_1$               | 220          | Duração do sinal de início dos processos do periférico enviado desde o <i>MicroBlaze</i> .   |
| $t_2$               | 633          | Tempo que leva obter o ganho de Kalman. conta desde o flanco de decida do sinal <code>En_grl</code> até o flanco de subida do sinal <code>GanhoSet</code> .  |
| $t_3$               | 175          | Tempo consumido desde o flanco de decida do sinal <code>dadoSet</code> até o sub-periférico <b>covari-<br/>anci&amp;Ganho</b> iniciar de novo seus cálculos uma vez o sinal <code>proxGanho</code> seja ativada.                         |
| $t_4$               | não definido | A duração de este tempo depende do programa no <i>MicroBlazee</i> quando o usuário decida iniciar o processo.  |
| $t_5$               | 220          | Duração do sinal que vem desde o <i>MicroBlaze</i> e inicia a estimação e a detecção de descontinuidades.  |
| $t_6$               | 152          | Tempo consumido pelo sub-periférico <b>estima-<br/>dor&amp;erro</b> desde o flanco de decida do sinal <code>dadoSet</code> para ativar o sinal <code>proxGanho</code> e iniciar os cálculos do periférico <b>covariancia&amp;Ganho</b> . |
| $t_7$               | 60           | Tempo que o sinal <code>proxGanho</code> esta ativa.   |
| $t_8$               | 313          | Tempo consumido pelo sub-periférico <b>estima-<br/>dor&amp;erro</b> desde o flanco de decida do sinal <code>dadoSet</code> nos seus cálculos até obter o dado do erro.   |
| $t_9$               | 40           | Tempo que o sinal <code>erroSet</code> esta ativa e inicia os cálculos do sub-periférico <b>CUSUM&amp;limiar</b> .   |
| $t_{10}$            | 850          | Tempo total consumido pelo periférico desde o flanco de decida do sinal <code>dadoSet</code> para obter o valor estimado e detetar alguma descontinuidade.   |
| $t_{11}$            | 60           | Tempo que o sinal <code>limirSet</code> esta ativa, indicando que este periférico terminou seu trabalho.   |

Na arquitetura da figura 4.10 mantém-se o nome dos sinais que foram usadas em cada periférico individual, além tem-se o sinal de entrada `iniciarEst` e o sinal de saída `saidaMono`. Uma vez o sinal `iniciarEst` é ativado inicia-se pela primeira vez o processo de fazer a estimativa do dado de entrada com a conversão análoga/digital. O sinal `iniciarEst` ativa-se depois que o sinal `En_grl` tinha sido ativada pelo *MicroBlaze* no início do software que roda em ele. O sinal `iniciarEst` foi conectado a uma porta OR para que as estimações depois da primeira e até a última sejam iniciadas pela ativação do sinal `limiarSet`.

Quando um alarme por descontinuidade é gerado, o sinal `alarm` permanece ativo apenas por 40 ns, mas para que o usuário consiga perceber ele tem desenvolvido o periférico **monoestavel**. O periférico **monoestavel** é um monoestável reativável cujo sinal de saída `saidaMono` permanece ativo por 1 s uma vez é ativado pelo sinal de alarme. A saída do monoestável esta mapeada a uns LEDs da placa de desenvolvimento para que o alarme possa ser percebido pelo usuário de maneira *on-line* durante o processo de soldagem.

O sinal de saída `alarm` será ligada no periférico que tem implementada a memoria que armazena os dados de onde as descontinuidades aconteceram.

Com o propósito de conhecer quanto tempo consome o periférico **conversaodetector**, foi feita a figura 4.11. Nesta figura apresenta-se a resposta no tempo do sinal `limiarSet` que inicia novamente a conversão análoga/digital e a estimativa.



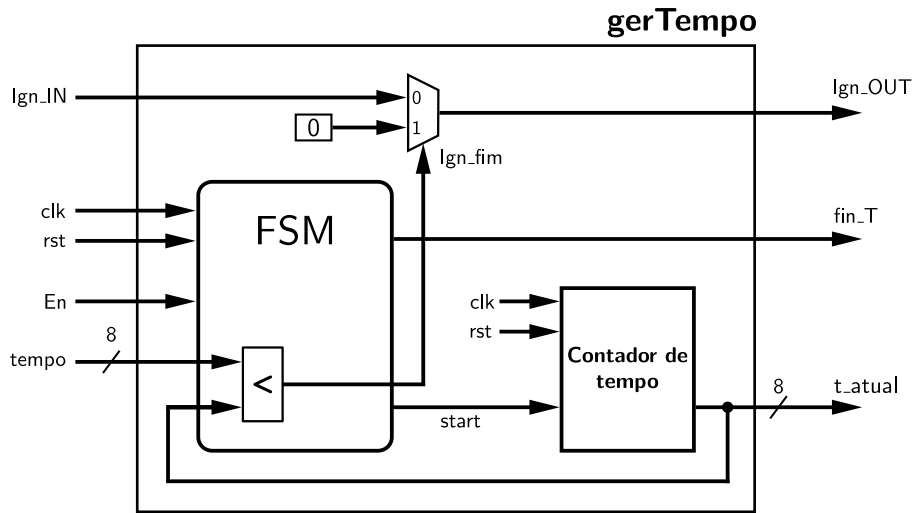
**Figura 4.11.** Diagrama de resposta no tempo para do sinal `limiarSet` que indica o tempo consumido desde o início da conversão análoga/digital até a estimativa.

O tempo consumido é contado depois da primeira estimativa, e considerando que o cálculo do ganho de Kalman está preparado para o uso. De esta maneira, o tempo total consumido na operação tem um período de  $4,32 \mu\text{s}$  com um ciclo útil de 60 ns, ou seja, uma frequência máxima de operação de 231,5 kHz.

#### 4.1.4 Gerenciador do tempo

O periférico que irá apresentar e descrever aqui controla o tempo do processo de soldagem e é chamado de **gerTempo**, o diagrama da sua arquitetura é apresentado na figura 4.12.

A FSM deste periférico controla o fluxo de dados do periférico e é iniciada quando o sinal de



**Figura 4.12.** Diagrama da arquitetura do periférico **gerTempo** que é ativado pelo sinal de entrada **En** e conta o tempo do processo de soldagem .

entrada **En** é ativado, o que acontece quando o arco é aberto. O que o periférico faz é receber do *MicroBlaze*, no sinal de entrada **tempo**, o valor do tempo que o usuário fixo para o processo de soldagem, e iniciar (**start** = 1) no bloco **Contador de tempo** uma contagem incremental de tempo em segundos até que a contagem atinja o valor de tempo do processo de soldagem e assim fechar o arco (**Ign\_fim**=1 e **start**=0). Quando o tempo é atingido a FSM também ativa o sinal de saída **fin\_T** que irá conectar ao gerenciador de interrupções.

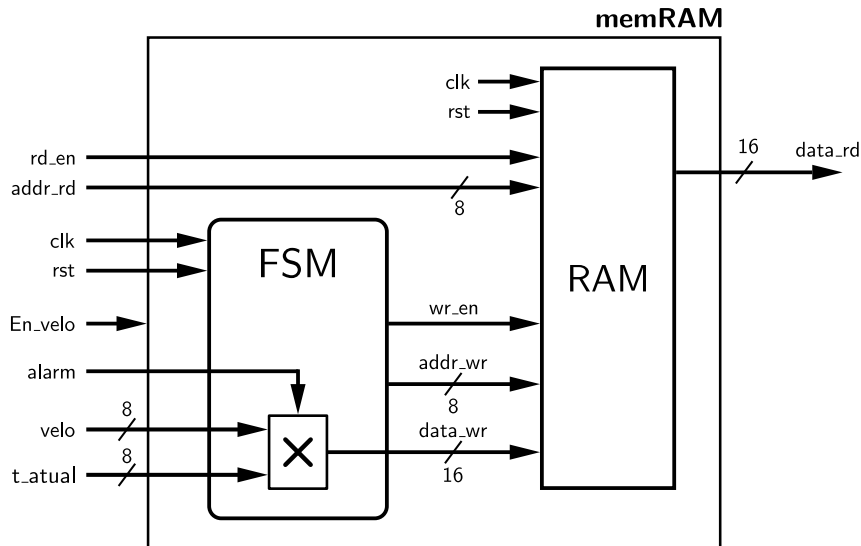
O valor da contagem incremental de tempo está sempre presente no sinal de saída **t\_atual** que será usado pelo periférico que possui implementada a memória das discontinuidades cada vez que um alarme aconteça.

Este periférico também recebe do *MicroBlaze* o sinal de ignição do arco, que sai do periférico pelo sinal **Ign\_OUT** e da placa de desenvolvimento por um dos pinos de entrada/saída de propósito geral, que está conectado a um *relay* e este a sua vez vai para a fonte de soldagem, como foi apresentado na figura 3.4.

#### 4.1.5 Memória de discontinuidades.

O periférico que a seguir se apresenta e descreve é sobretudo uma memória RAM chamado de **memRAM**, o diagrama da arquitetura deste periférico é apresentado na figura 4.13.

A FSM do periférico recebe do *MicroBlaze*, no sinal de entrada **velo**, o valor da velocidade do processo de soldagem escolhido pelo usuário e inicia uma vez o sinal de entrada **En\_velo** é ativado, também pelo *MicroBlaze*. Quando uma discontinuidade é encontrada pelo periférico **conversaodetector**, o sinal de entrada **alarm** é ativado fazendo que a FSM do **memRAM** acu-



**Figura 4.13.** Diagrama da arquitetura do periférico **memRAM** que armazena a quantidade de descontinuidades e onde elas aconteceram.

mule o número de descontinuidades e tome os sinais de entrada **velo** e **t\_atual** e os multiplique, para obter a posição da descontinuidade. Depois da multiplicação a FSM escreve na memória RAM primeiro o acumulado das descontinuidades encontradas e segundo a posição da última descontinuidade encontrada.

Na memória RAM armazenam-se a quantidade de descontinuidades encontradas durante o processo de soldagem e a posição (em milímetros) onde estas aconteceram. A posição na memória para a quantidade de descontinuidades é reescrita cada vez que uma descontinuidade é encontrada e armazenada. Esta memória RAM só pode ser lida pelo *MicroBlaze* uma vez terminado o processo de soldagem. A posição da descontinuidade armazenada na memória é marcada relativa ao ponto de início do processo de soldagem.

#### 4.1.6 Gerenciamento do *MicroBlaze*.

Os periféricos desenvolvidos e, até aqui descritos não conseguiram detectar as descontinuidades no processo de soldagem sem ter um processador que os gerencie. Neste contexto é incluído o processador *soft* da Xilinx: o *MicroBlaze*, cuja função é gerenciar e intercambiar dados entre os periféricos do sistema de detecção de descontinuidades. No entanto, foi preciso adicionar outros periféricos de propósito geral que estão associados ao hardware presente na placa de desenvolvimento [33], e que foram usados principalmente para as tarefas da interface de usuário.

Antes de apresentar toda a arquitetura do sistema de detecção de descontinuidades, no qual inclui o *MicroBlaze* e todos os periféricos envolvidos, serão apresentados e descritos os periféricos de propósito geral adicionados, que podem ser incluídos dentro dos seguintes itens:

- Entrada/saída de propósito geral.
- RS-232 serial.
- RS-232 modem.
- LCD.
- *Timer*.
- Interrupções.

#### 4.1.6.1 Entrada/saída de propósito geral

Este item inclui os periféricos da placa: os botões e as chaves. Os botões da placa são utilizados para escolher opções que estão apresentadas no LCD e são gerenciadas pelo programa que roda no *MicroBlaze*. As opções definem os parâmetros do processo de soldagem como velocidade da mesa posicionadora e o tempo do processo de soldagem. As chaves usam-se no momento de enviar a informação das descontinuidades detetadas para o PC.

#### 4.1.6.2 RS-232 serial

Este periférico usa a porta serial da placa de desenvolvimento identificado como DCE [33], que também pode ser identificado por ser a porta serial fêmea das duas disponíveis na placa. O periférico foi utilizado com mais frequência na etapa de depuração, mas no sistema de detecção das descontinuidades foi utilizado para enviar ao PC a informação das descontinuidades encontradas durante o processo de soldagem.

#### 4.1.6.3 RS-232 modem

Este periférico utiliza a porta serial da placa de desenvolvimento identificado como DTE [33], que é a porta serial macho na placa. O periférico foi utilizado para enviar ao controlador da mesa posicionadora da bancada de soldagem, os parâmetros com o quais ela vai trabalhar. Os parâmetros enviados por esta porta são: a velocidade de deslocamento da mesa, o tempo do processo e a direção de deslocamento. O comando para iniciar o deslocamento da mesa também é enviado por esta porta, mas isto só acontece depois que o sistema está pronto para soldar e o arco é aberto, lembrando que isto é gerenciado pelo programa que roda no *MicroBlaze*.

#### 4.1.6.4 LCD

O LCD da placa de desenvolvimento é o meio em que são apresentados continuamente para o usuário as informações do sistema, do mesmo modo em que são apresentadas as escolhas para os parâmetros de soldagem. Também inclui-se, uma vez finalizado o processo, a apresentação da informação das descontinuidades que foram encontradas.

#### 4.1.6.5 *Timer*

O *Timer* é um periférico usado para gerar atrasos na execução do programa escrito em C que roda no *MicroBlaze*.

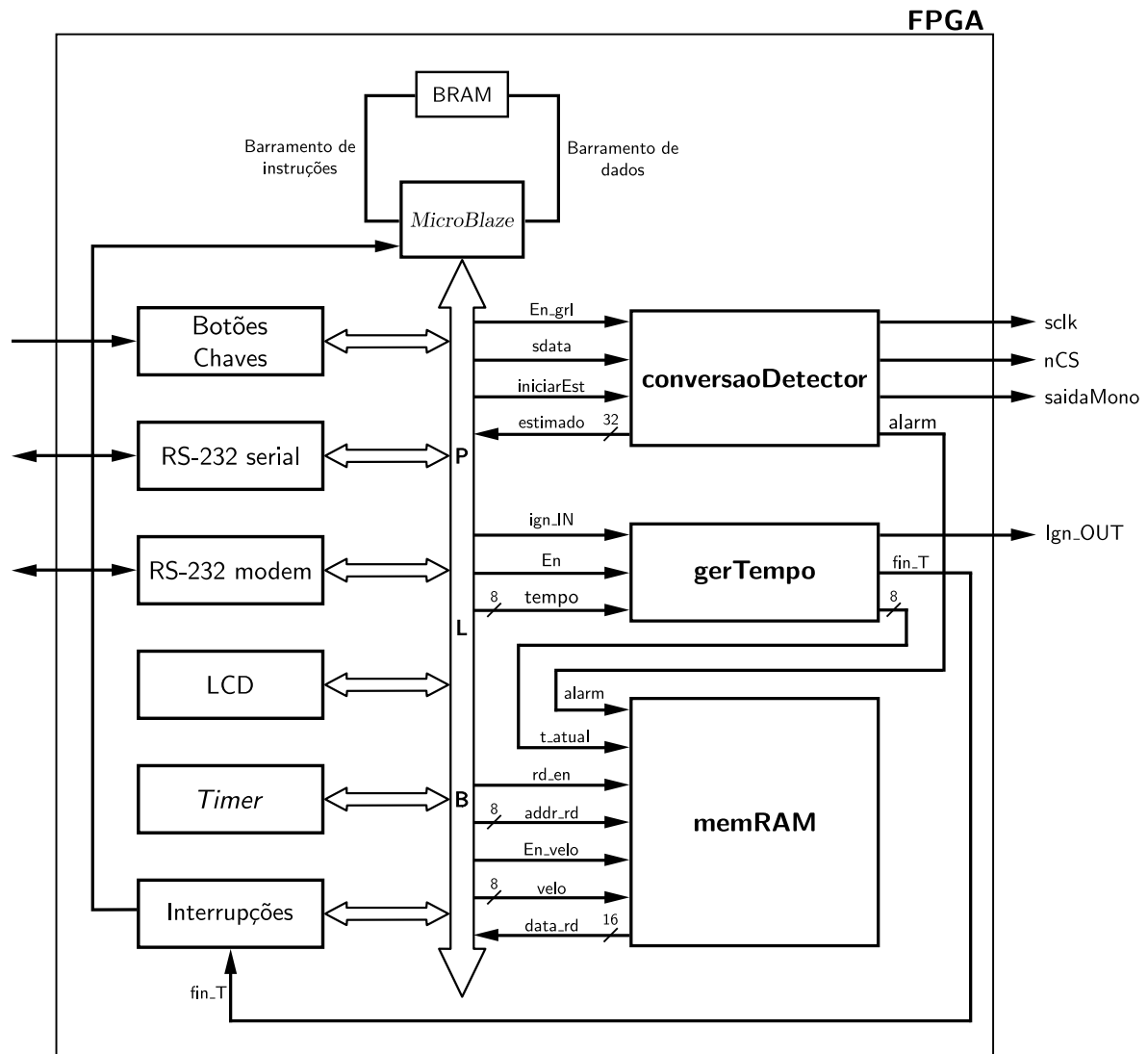
#### 4.1.6.6 Interrupções

Este periférico gerencia as interrupções do sistema de detecção das descontinuidades, uma vez o serviço de interrupções do periférico é configurado no programa que roda no *MicroBlaze*. Este periférico está ligado ao hardware do processador *MicroBlaze* e ao periférico **gerTempo**. Quando o **gerTempo** termina a sua conta, o sinal `fim_T` ativa o periférico e a interrupção é atendida.

Expondo todos os periféricos do sistema de detecção das descontinuidades, a figura 4.14 apresenta a maneira como foram conectados para conformar a arquitetura do sistema. A figura também inclui um bloco de memória SRAM que contém o programa desenvolvido em C para o gerenciamento do sistema.

A figura 4.14 mostra que o barramento PLB (*Processor Local Bus*) [27], que é criado pela mesma ferramenta software que cria o *MicroBlaze*, foi escolhido para interconectar o *MicroBlaze* com os periféricos.





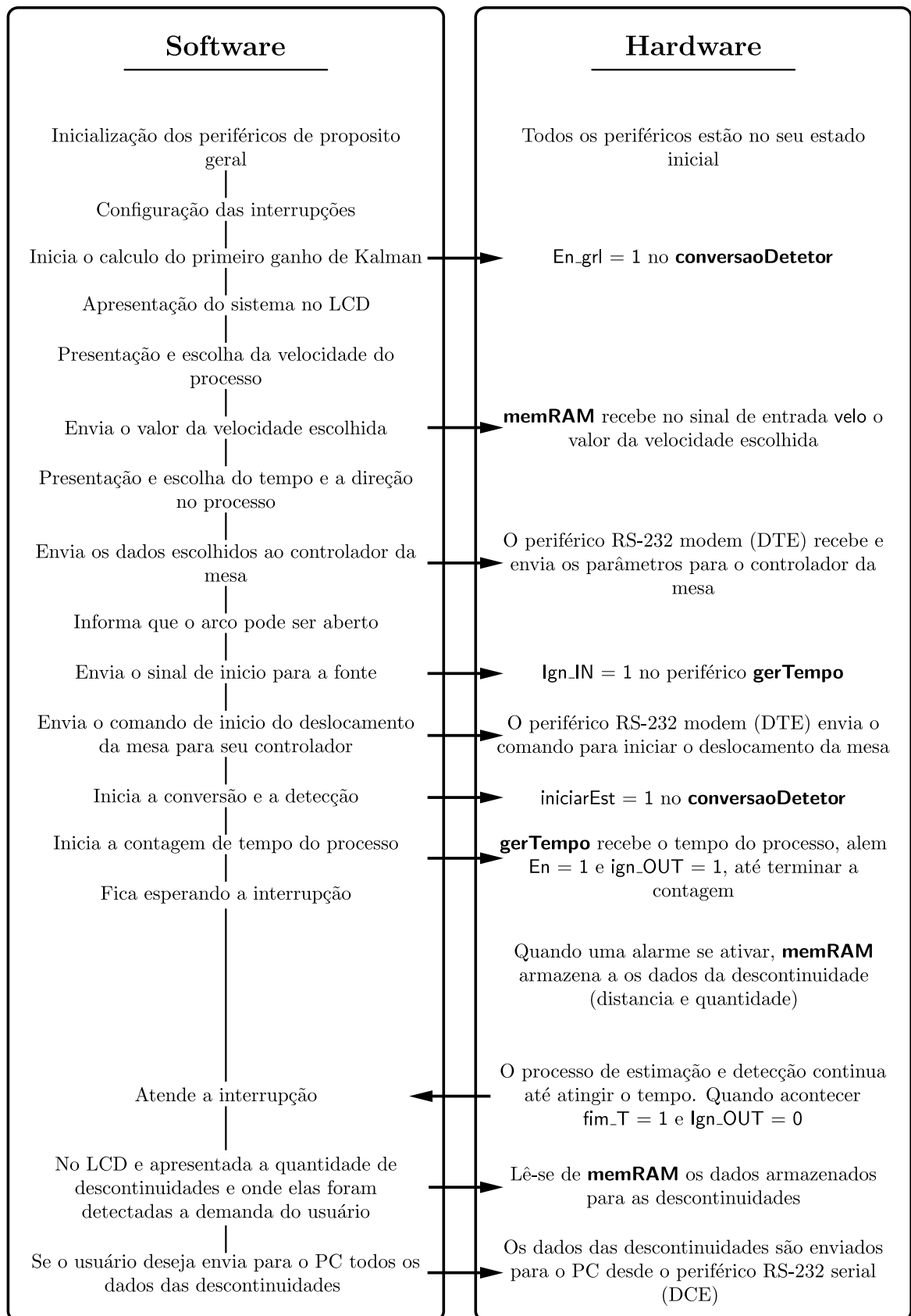
**Figura 4.14.** Diagrama da arquitetura do sistema de detecção de descontinuidades que apresenta o *MicroBlaze* e os periféricos envolvidos.

#### 4.1.7 Fluxo de execução

A figura 4.15 apresenta um diagrama para explicar como é o fluxo de execução na detecção de descontinuidades no processo de soldagem, como complemento na descrição nos periféricos apresentados anteriormente. Apresentam-se a parte software que roda seqüencialmente no *MicroBlaze* e a parte hardware dos periféricos personalizados rodando paralelamente, as setas indicam quando desde o programa seqüencial são ativados os periféricos na parte hardware.

#### 4.1.8 Consumos de recursos.

A seguir encontra-se os resultados dos recursos consumidos do FPGA. Para isto é preciso lembrar que o FPGA usado foi a Spartan-3E 1600 da Xilinx [38] com um relógio de 50 MHz.



**Figura 4.15.** Fluxo de execução na detecção de descontinuidades. Estão incluído o software e o hardware. As execuções no hardware são concorrentes.

A apresentação do consumo de recursos está contido nas seguintes tabelas. A tabela 4.3 tem consignada o consumo de recursos para o *MicroBlaze* e os periféricos de propósito geral (botões, LCD, *Timer*, etc.), na tabela 4.4 esta consignado o consumo de recursos para os periféricos personalizados desenvolvidos.

**Tabela 4.3.** Relação de consumo de recursos para o *MicroBlaze* e os periféricos de propósito geral usados.

| Dispositivo              | Parâmetros  |                  |              |           |
|--------------------------|-------------|------------------|--------------|-----------|
|                          | Slices      | Slices Flip Flop | LUT          | MUL18x18  |
| <i>MicroBlaze</i>        | 565 (3,83%) | 711 (2,41%)      | 1202 (4,07%) | 3 (8,33%) |
| Periféricos predefinidos | 850 (5,76%) | 1108 (3,76%)     | 935 (3,17%)  | 0 (0%)    |

**Tabela 4.4.** Relação de consumo de recursos para os periféricos personalizados.

| Dispositivo              | Parâmetros    |                  |               |             |
|--------------------------|---------------|------------------|---------------|-------------|
|                          | Slices        | Slices Flip Flop | LUT           | MUL18x18    |
| <b>conversaoDetector</b> | 4652 (31,54%) | 1949 (6,61%)     | 8878 (30,09%) | 24 (66,67%) |
| <b>gerTempo</b>          | 384 (2,60%)   | 506 (1,72%)      | 608 (2,06%)   | 0 (0%)      |
| <b>memRAM</b>            | 358 (2,43%)   | 489 (1,66%)      | 545 (1,85%)   | 1 (2,78%)   |

No processo de criação o processador *Soft MicroBlaze*, a ferramenta software usada cria mais outros módulos hardware para a operação ótima do processador, como os *Buses*, *Memória*, *controladores de memória*, *gerenciador de relógio* e o *gerenciador de reset*, o consumo de recursos de hardware para estes módulos apresenta-se na tabela 4.5.

**Tabela 4.5.** Relação de consumo recursos para os dispositivos extras usados pelo *MicroBlaze*.

| Dispositivo         | Parâmetros  |                  |             |          |
|---------------------|-------------|------------------|-------------|----------|
|                     | Slices      | Slices Flip Flop | LUT         | MUL18x18 |
| Outras entidades HW | 363 (2,46%) | 241 (0,82%)      | 598 (2,03%) | 0 (0%)   |

O total dos recursos consumidos, ou seja, a soma dos recursos consumidos nas tabelas 4.3, 4.4 e 4.5, é apresentado na tabela 4.6.

**Tabela 4.6.** Consumo total de recursos do FPGA.

| Parâmetro        | Usado | Disponível | Porcentagem |
|------------------|-------|------------|-------------|
| Slices           | 7172  | 14752      | 48,62       |
| Slices Flip Flop | 5004  | 29504      | 16,96       |
| LUT              | 12766 | 29504      | 43,27       |
| MUL18x18         | 28    | 36         | 77,78       |

Os resultados para a análise de tempo, feito pelo software XPS *Xilinx Platform Studio* (EDK 10.1), são apresentados na tabela 4.7<sup>1</sup>. São apresentados os resultados de frequência para *MicroBlaze*, os periféricos personalizados e a menor frequência tanto para os periféricos de propósito geral como para as outras entidades hardware.

**Tabela 4.7.** Máximas frequências de operação calculadas.

| Dispositivo                    | Frequência Máxima (MHz) |
|--------------------------------|-------------------------|
| <i>MicroBlaze</i>              | 95.497                  |
| <b>conversaoDetector</b>       | 38.688                  |
| <b>gerTempo</b>                | 101.740                 |
| <b>memRAM</b>                  | 102.093                 |
| Periféricos de propósito geral | 128.287                 |
| Outras entidades hardware      | 164.096                 |

## 4.2 RESULTADOS DE SOLDAGEM.

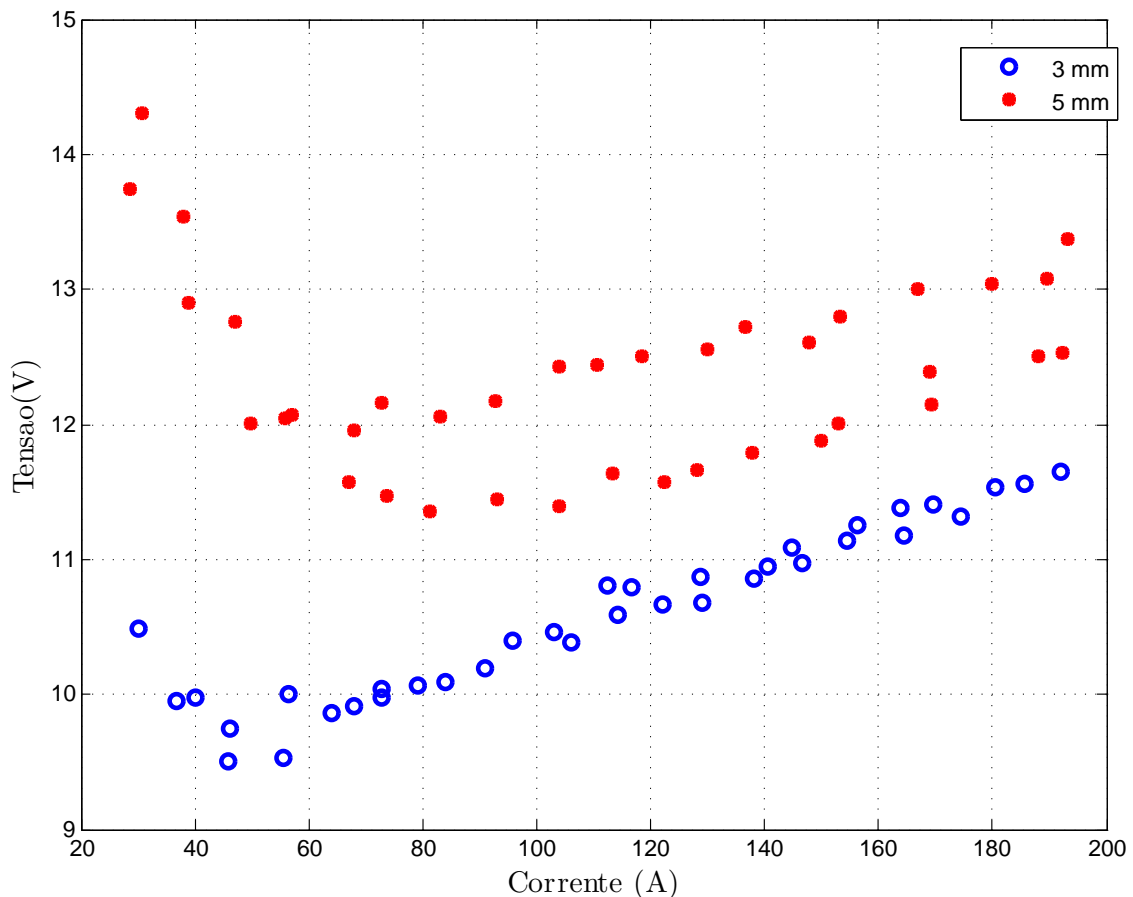
A seguinte parte na apresentação de resultados está relacionados com a soldagem, ou seja, a validação do sistema de detecção de descontinuidades em um processo de soldagem GTAW. Nos resultados de soldagem incluem-se o resultado da caracterização do arco GTAW, os cordões de solda para obter o padrão e os cordões com descontinuidades intencionais que foram analisados off-line para ajustar o algoritmo de detecção de mudança. Também apresentam-se os resultados usando o sistema de detecção de descontinuidades embarcado no FPGA.

<sup>1</sup>Os valores reportados aqui são estimativas pós-sínteses e calculados para cada periférico. Estes valores mudarão depois do que *Place and Route* seja feito para o sistema (FPGA).

### 4.2.1 Curva característica do sistema de soldagem.

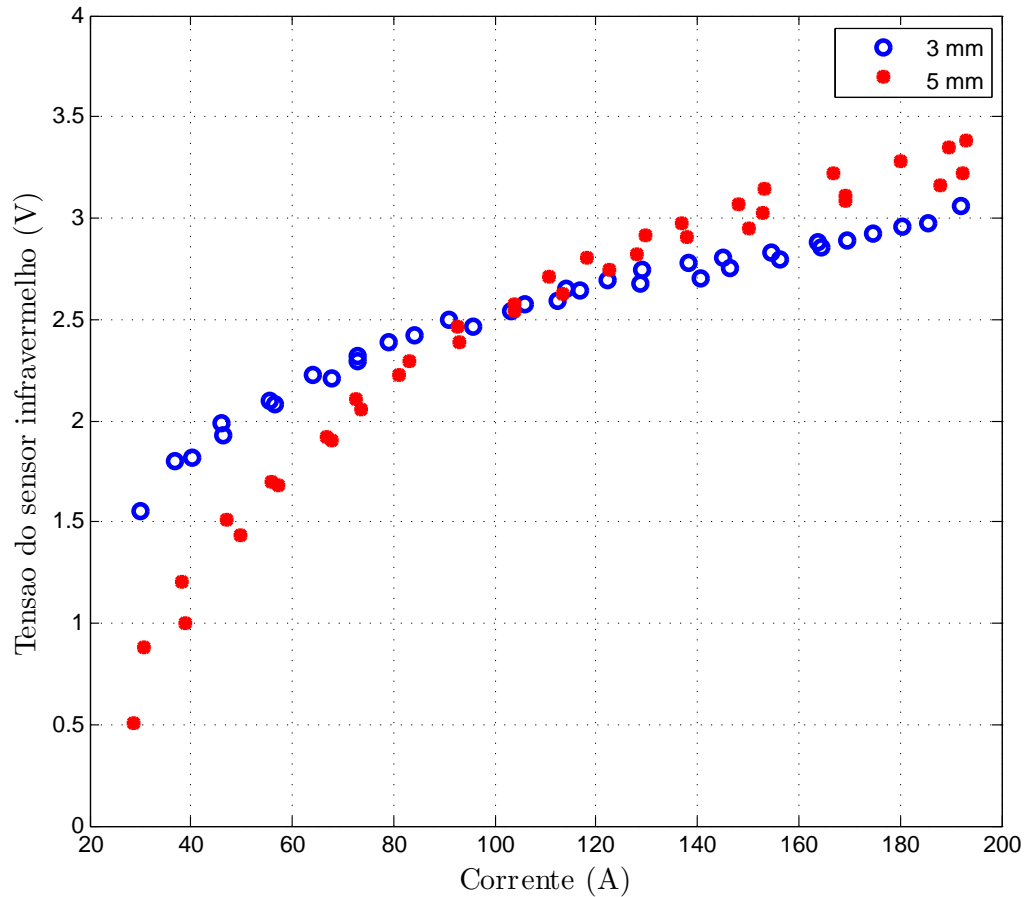
A curva característica do arco GTAW, que é uma curva tensão-corrente, foi feita usando a bancada apresenta na figura 3.6 e seguindo os passos descritos nada seção 3.2.1.1. A faixa de corrente escolhida para a experiência foi de 30 até 200 A. Uma vez o arco foi aberto os valores de corrente aumentaram progressivamente de 10 A. Para coletar os dados e levantar a curva foi desenvolvido um programa de aquisição de dados (tensão do arco, corrente do arco e tensão do sensor de infravermelho) no software LabVIEW.

Desta maneira foram obtidas as curvas características. A figura 4.16 apresenta a curva característica do arco GTAW para a fonte de soldagem usada, a figura 4.17 apresenta a curva do sensor de infravermelho para a experiência da caracterização do arco. Nestas curvas os pontos representados por estrelas correspondem a um *stand-off* de 5 mm em quanto os pontos representados por círculos correspondem a um *stand-off* de 3 mm.



**Figura 4.16.** Curva característica do arco GTAW. A faixa de corrente foi 30-200 A e o *stand-off* foi 3 e 5 mm.

A figura 4.16 mostra que a região estável inicia aproximadamente a os70 A. Segundo este resultado foi escolhida a região entre os 80 e 180 A, para obter o padrão de solda do sistema de detecção de mudança.



**Figura 4.17.** Curva de resposta do sensor de infravermelho para a caracterização do arco GTAW, com uma faixa de corrente de 30-200 A e *stand-off* de 3 e 5 mm.

Na figura 4.17 observa-se o comportamento esperado para um sistema térmico (sistema de primeiro ordem), o valor de tensão do sensor de infravermelho aumenta exponencialmente quanto a corrente de arco é aumentada. Também pode-se ver que inicialmente a taxa de incremento na tensão do sensor é maior para o comprimento de *stand-off* de 5 mm do que para o caso quando o *stand-off* é 3 mm, até aproximadamente os 100 A em que a taxa de incremento diminui.

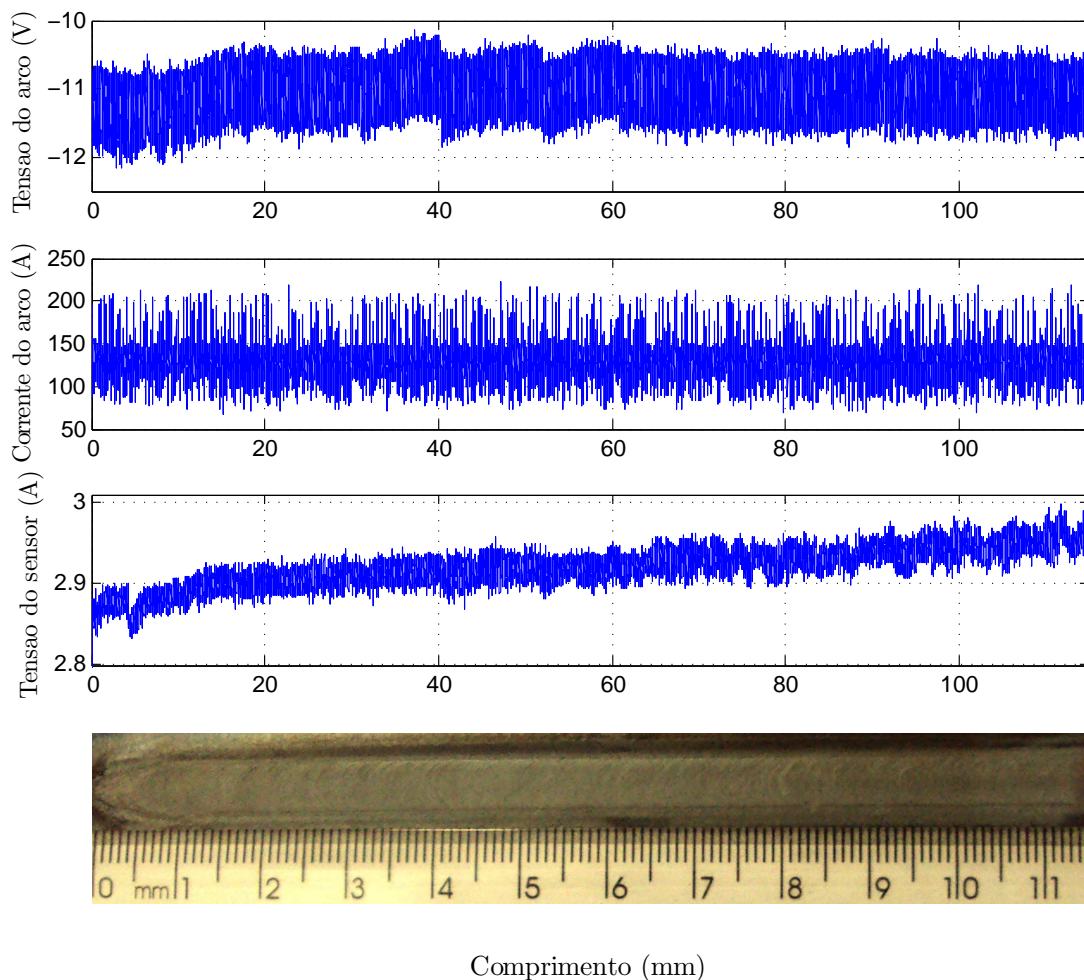
## 4.2.2 Ajuste do sistema de detecção de mudanças

Como foi apresentado previamente é preciso fazer um ajuste do sistema de detecção de mudanças. Para este ajuste primeiro obtêm-se um padrão de solda, um cordão sem discontinuidades aparentes. A seguinte parte no ajuste do sistema de detecção de discontinuidades é fazer alguns

testes de soldagem com descontinuidades intencionais para ajustar de maneira *off-line* tanto o valor de  $Q$ ,  $\nu$  e o limiar  $h$ . Os resultados obtidos do ajuste serão apresentados a seguir.

#### 4.2.2.1 Cordão padrão de solda.

O padrão de solda é um cordão de solda sem descontinuidades aparentes, como o apresentado na figura 4.18, este cordão foi obtido com uma corrente de 140 A, *stand-off* de 5 mm e uma velocidade de deslocamento de 2,5 mm/s. Além da figura apresentar o aspecto do cordão, mostra as curvas de tensão do arco, corrente do arco e a tensão do sensor infravermelho, que é o sinal de interesse.



**Figura 4.18.** Cordão de solda padrão sem descontinuidades aparentes que foi conseguido a 140 A, e com um *stand-off* de 5 mm. De acima para abaixo na figura estão a tensão do arco, a corrente do arco, a tensão do sensor infravermelho e o aspecto que o cordão tem.

Para ter um padrão melhor definido foram feitos vários cordões sem a presença aparente de descontinuidades, nos que possuíam a mesma vazão de gás (12 L/min) e a mesma velocidade de deslocamento (2,5 mm/s), mudavam apenas o *stand-off* e a corrente do arco. Os valores de corrente foram 100, 140 e 180 A, que estão entre os extremos e a média da região estável do arco. A tabela 4.8 apresenta a resposta do sensor infravermelho para a combinação de parâmetros corrente-*stand-off* em cada cordão de solda. Também há um teste a mais a corrente de 140 A, isto é por ser está a corrente escolhida para fazer os testes de detecção de descontinuidades.

**Tabela 4.8.** Resposta do sensor infravermelho no processo de obter o padrão de solda, os parâmetros corrente e *stand-off* são mudados, em quanto a vazão de gás e velocidade permanecem constantes.

| Corrente (A) | <i>stand-off</i> |      |
|--------------|------------------|------|
|              | 3 mm             | 5 mm |
| 100          |                  |      |
| 140          |                  |      |
| 140          |                  |      |
| 180          |                  |      |

Como corresponde, de cada curva de tensão do sensor infravermelho na tabela 4.8 é obtido o valor da variância  $R$  para usar no algoritmo de detecção de mudança. Os resultados de  $R$  para cada curva são apresentados na tabela 4.9, a qual obtêm que o valor médio da variância  $R = 0.0029$ .



**Tabela 4.9.** Valores da variância para cada cordão sem descontinuidades usados para obter o padrão. O valor médio para estes dados é  $R = 0.0029$

| Corrente (A)          | 100    |        | 140    |        |        |        | 180    |        |
|-----------------------|--------|--------|--------|--------|--------|--------|--------|--------|
| <i>Stand-off</i> (mm) | 3      | 5      | 3      | 5      | 3      | 5      | 3      | 5      |
| $R$                   | 0.0022 | 0.0016 | 0.0029 | 0.0011 | 0.0082 | 0.0006 | 0.0021 | 0.0047 |

Antes de apresentar os cordões com descontinuidades e usar o algoritmo de detecção de mudança neles, é necessário saber se o sinal de interesse tem uma distribuição normal como é exigido pelo modelo 2.1. Para a comprovação foram levantadas as curvas de probabilidade normal com os resíduos dos dados da tabela 4.8 obtidos na corrente de interesse (140 A) dentro do nível de significância de 95%, na figura 4.19 apresentam-se estas curvas. Para que os valores dos resíduos obedeçam à distribuição normal, o valor de probabilidade  $p$  deverá estar acima de 0,005.

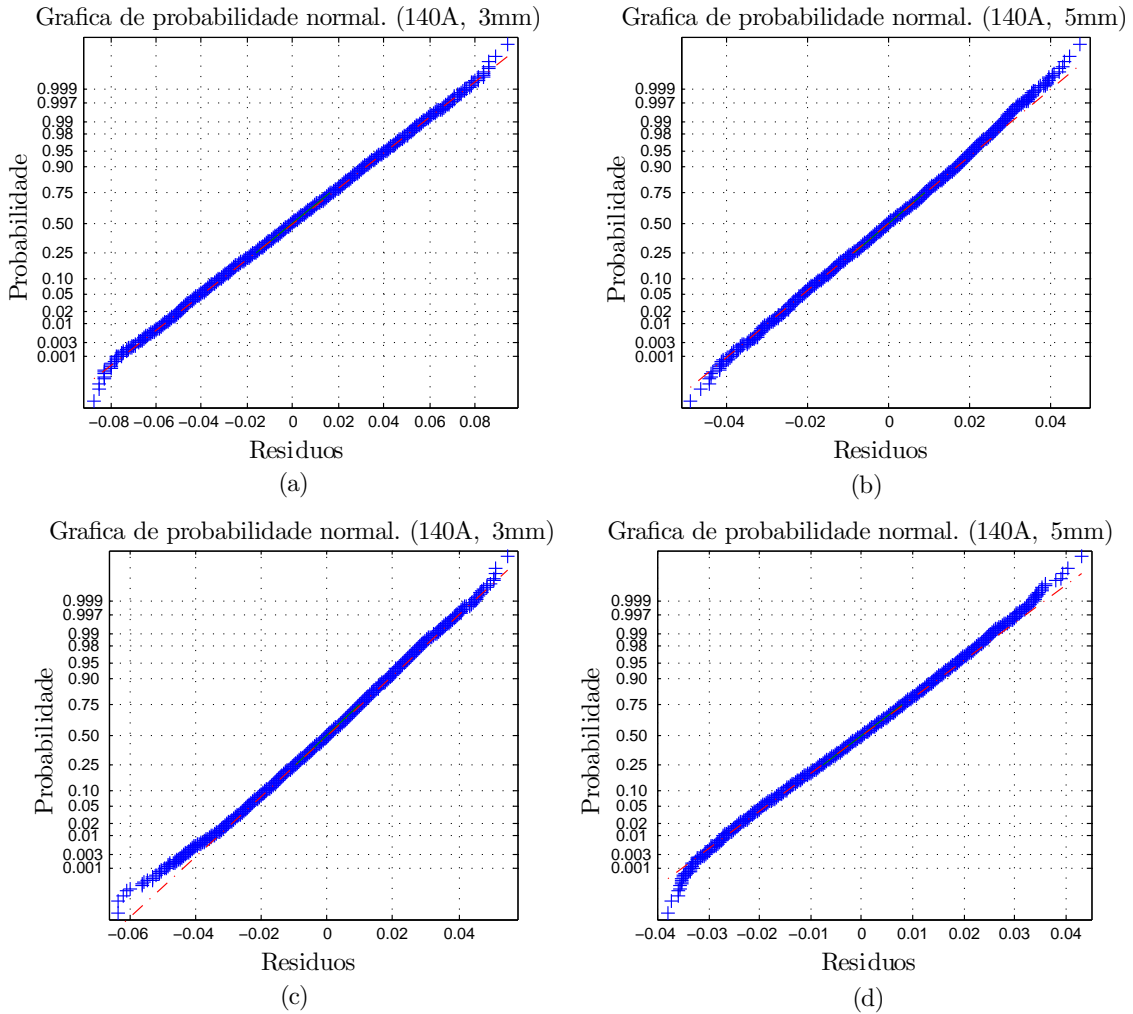
Das curvas da figura 4.19 e os valores de  $p$  maiores do que 0,005 pode-se afirmar que os dados seguem uma distribuição normal e pode-se utilizar a técnica de detecção de mudança apresentado na seção 2.4.

#### 4.2.2.2 Cordões de soldagem com descontinuidades.

Os cordões de soldagem que a continuação se apresentam contém descontinuidades introduzidas intencionalmente, com o propósito de ajustar os valores de  $Q$ ,  $\nu$  e o limiar  $h$ . O tipo de descontinuidades e os parâmetros de soldagem usados nestes cordões foram os apresentados na seção 3.3. Para as descontinuidades por arame usa-se a expressão *par distância/quantidade-de-arames*, para indicar a que distância relacionadas ao início da solda foram a locadas as descontinuidades e que quantidade de arame foi usada, por exemplo, se para 50 mm foram usadas 3 arames o par distância/quantidade-de-arames é: 50 mm/3.

Antes de continuar é bom ter uma idéia de como muda a resposta dos sinais em um cordão de solda com descontinuidades respeito do padrão. De esta maneira apresenta-se a figura 4.20, nesta figura tem-se um cordão com descontinuidades produzidas por furos, além o sinal de tensão do arco, a corrente do arco e o sinal de tensão do sensor infravermelho.

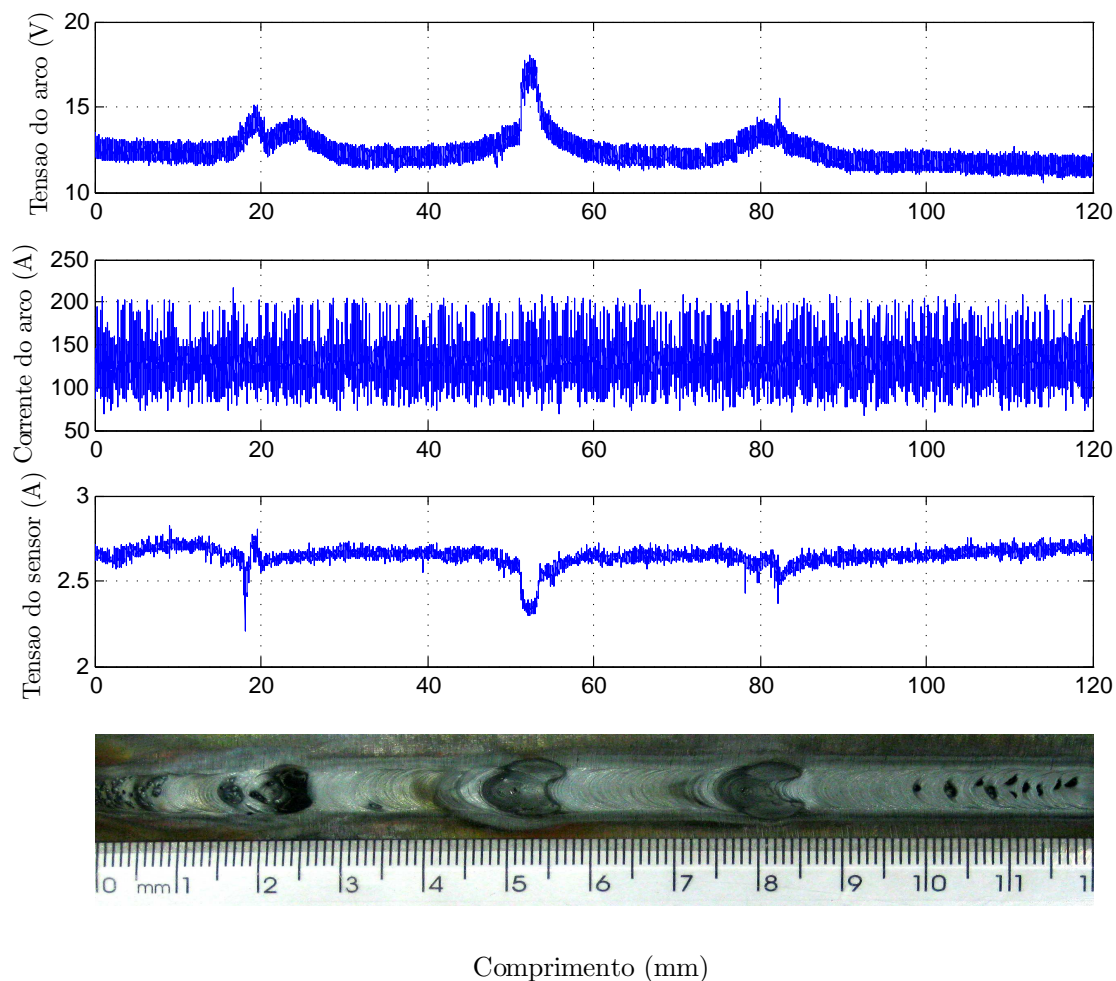
Na figura 4.20 é possível ver como a fonte mantém a corrente constante aumentando a tensão do arco quando apresenta uma variação no seu comprimento, para o caso, ocasionada pelo furo na peça. Na curva de tensão do sensor infravermelho também pode-se ver a perturbação no sinal produzida pela descontinuidade, são estas perturbações no sinal que se esperam detectar para gerar o alarme. Pode-se perceber que ainda sendo o mesmo tipo de descontinuidade a forma



**Figura 4.19.** Teste de normalidade dos resíduos do sinal do sensor infravermelho. (a) 140 A,  $stand-off = 3$  mm,  $p = 0,99999$  (b) 140 A,  $stand-off = 5$  mm,  $p = 0,99999$  (c) 140 A,  $stand-off = 3$  mm,  $p = 0,99998$  (d) 140 A,  $stand-off = 5$  mm,  $p = 0,99999$

da perturbação é diferente, é por esta razão que se faz o ajuste dos parâmetros do sistema de detecção de mudança e conseguir detetar a maior quantidade de perturbações no sinal.

O ajuste do algoritmo para detectar as mudanças com o sinal do sensor infravermelho consiste em dar valores iniciais a  $Q$ ,  $\nu$  e o limiar  $h$  e com estes fazer testes *off-line* do algoritmo para cada conjunto de dados por cordão, o software que se usou para isto foi o MatLab. Segundo o desempenho do algoritmo por cada conjunto de dados, os valores  $Q$ ,  $\nu$  e  $h$  podem mudar ou não, até conseguir os valores que dêem um senso favorável na detecção das discontinuidades. Neste contexto, o valor usado para a variância foi  $Q = 1 \times 10^{-6}$ , este baixo valor foi escolhido porque se deseja que o erro introduzido pelo modelo da equação (2.1) seja menor, ou seja, atribuí-se uma escolha favorável do modelo. Para o compensador inicialmente foi escolhida como valor a média da diferença entre dados consecutivos, mas finalmente utilizou-se o valor de  $\nu = 3 \times 10^{-5}$ .



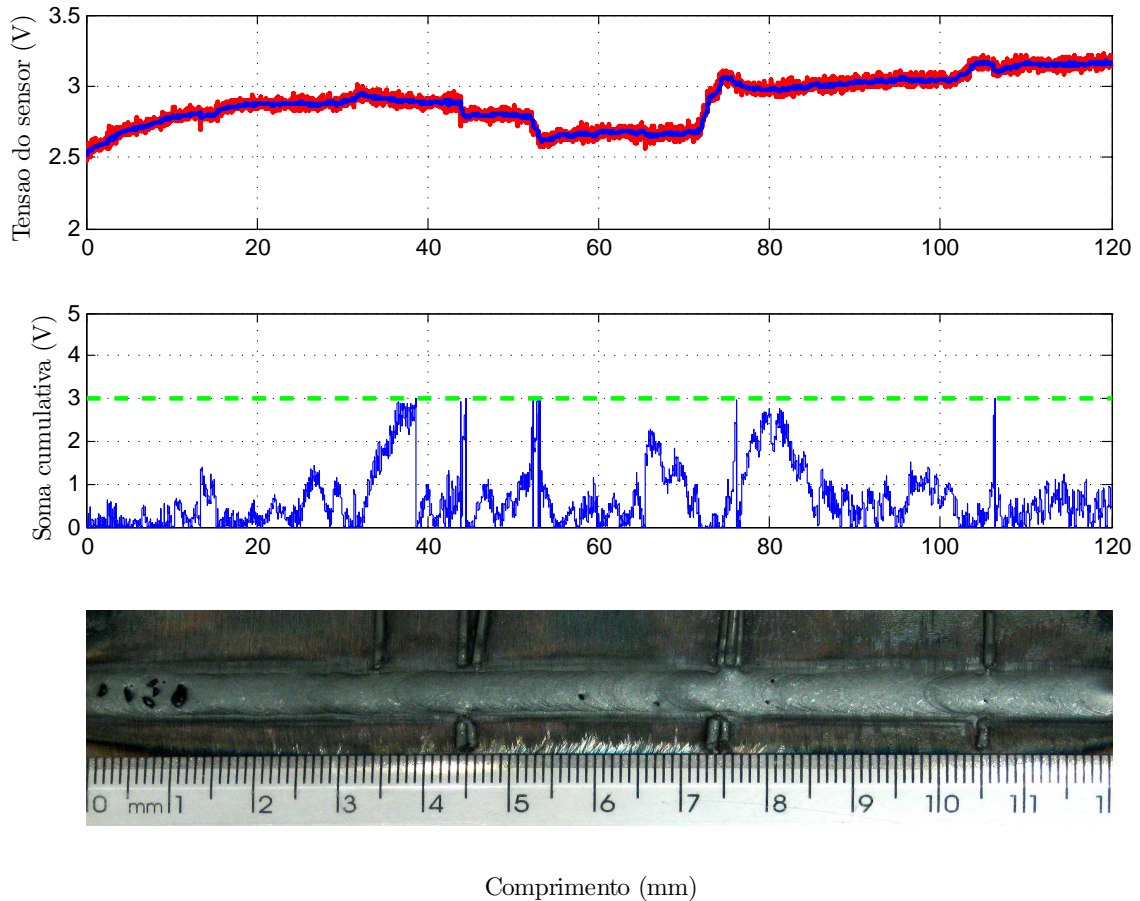
**Figura 4.20.** Cordão de solda com discontinuidades produzidas por furos, conseguido a 140 A e com um *stand-off* de 5 mm. De acima para abaixo na figura estão a tensão do arco, a corrente do arco, a tensão do sensor infravermelho e o aspecto que o cordão tem.

O valor do limiar  $h$  inicialmente é escolhido grande e é ajustado segundo tenha-se sucesso na detecção das discontinuidades, o valor do limiar usado finalmente foi  $h = 3$ .

A seguir apresentam-se alguns exemplos do resultado do ajuste na detecção de cada uma das discontinuidades propostas (arames, furos e areia), os cordões foram feitos com os parâmetros de soldagem consignados na tabela 3.3. Por cada exemplo se tem uma figura que na parte (a) tem o sinal do sensor infravermelho junto com seu valor estimado pelo filtro Kalman, na parte (b) tem o teste de hipóteses (a soma cumulativa e o limiar) e na parte (c) tem uma fotografia do cordão.

## Descontinuidades por arame

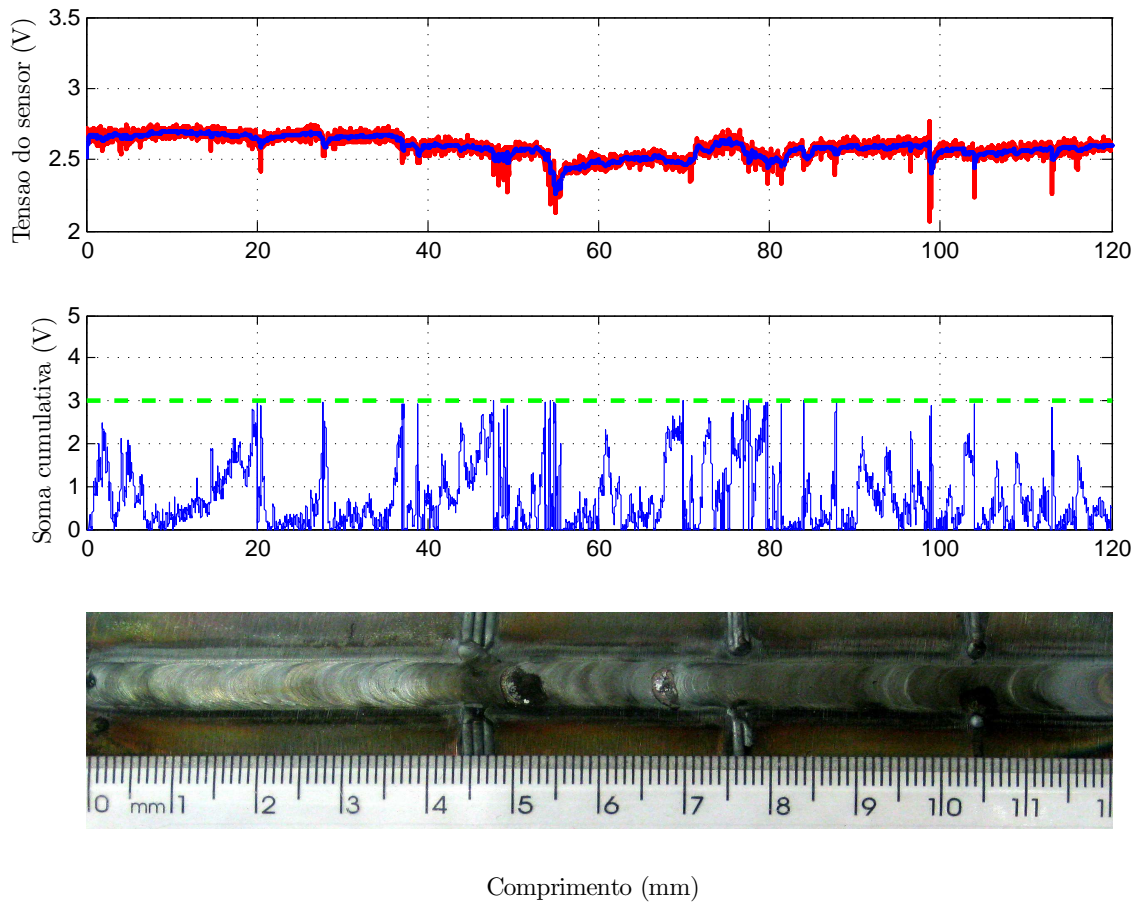
A figura 4.21 apresenta o resultado de um teste feito com um *stand-off* de 3 mm e com descontinuidades alocadas, relacionadas ao início da soldagem, aproximadamente nos seguintes par distância/quantidade-de-arame: i) 35 mm/1, ii) 45 mm/2, iii) 75 mm/3 e iv) 105 mm/1.



**Figura 4.21.** Resultado do ajuste para a detecção de descontinuidades por arame. *stand-off* = 3 mm e descontinuidades no par distância/quantidade-de-arame: 35 mm/1, 45 mm/2, 75 mm/3 e 105 mm/1.

Para esta experiência detetou-se o 75% das descontinuidades, só a descontinuidade 'i)' não é detetada, a perturbação que esta descontinuidade gera no sinal do sensor infravermelho resulta não significativa para o algoritmo de detecção.

A figura 4.22 apresenta o resultado de um teste feito com um *stand-off* de 5 mm e com descontinuidades alocadas, relacionadas ao início da soldagem, aproximadamente nos seguintes par distância/quantidade-de-aramé: i) 45 mm/3, ii) 75 mm/3 e iii) 105 mm/1.



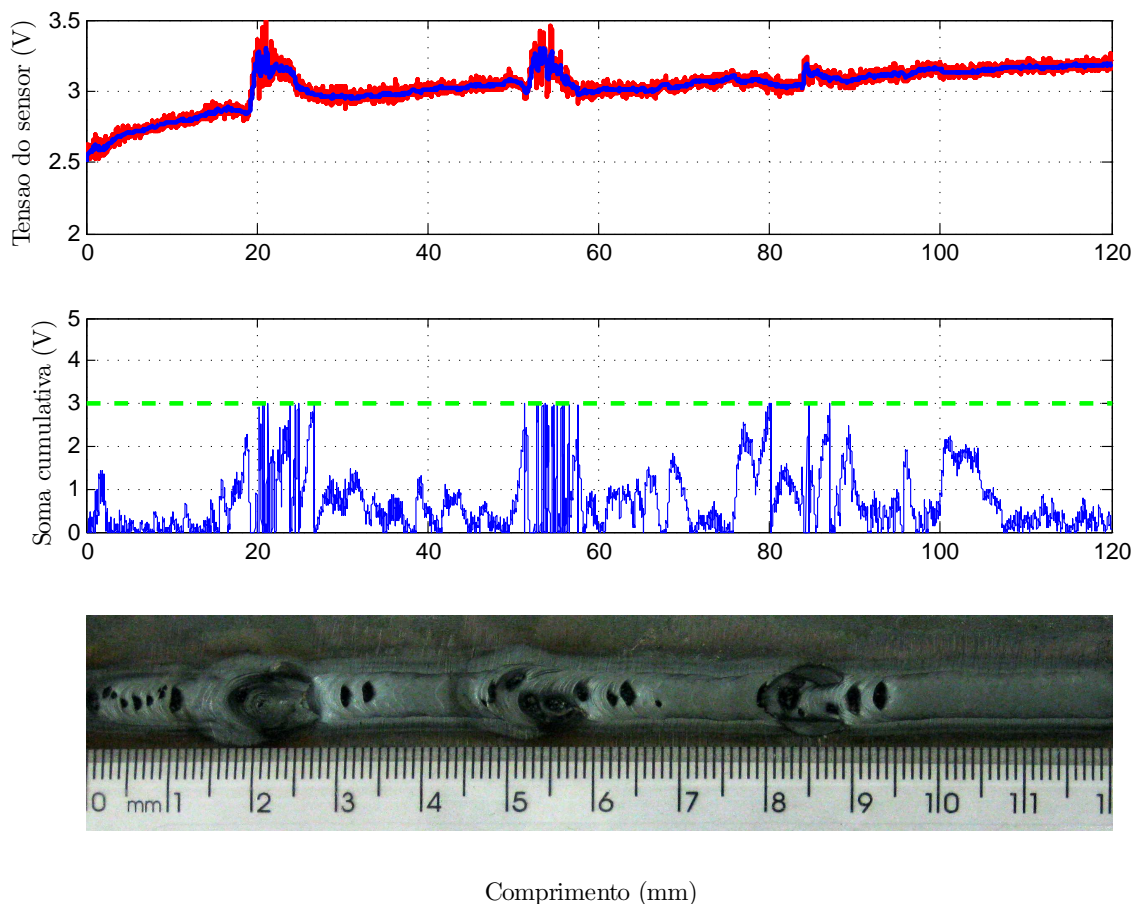
**Figura 4.22.** Resultado do ajuste para a detecção de descontinuidades por arame. *stand-off* = 5 mm e descontinuidades no par distância/quantidade-de-aramé: 45 mm/3, 75 mm/3 e 105 mm/1.

Nesta experiência são detetadas o 100% das descontinuidades e alguns alarmes falso, mas também tem-se alarmes falso. Alguns dos alarmes nesta experiência foram gerados por deslocamento de massa (protuberâncias) da descontinuidade mesma, como percebe-se nas descontinuidades 'i)' e 'ii)'.

Nas figuras 4.21 e 4.22 pode-se observar que para o *stand-off* de 3 mm se tem um sinal do sensor infravermelho mais suave do que com o *stand-off* de 5 mm, o que permite ter um resultado da soma cumulativa mais estável tendo como consequência uma menor detecção de alarmes falso.

## Descontinuidades por furos

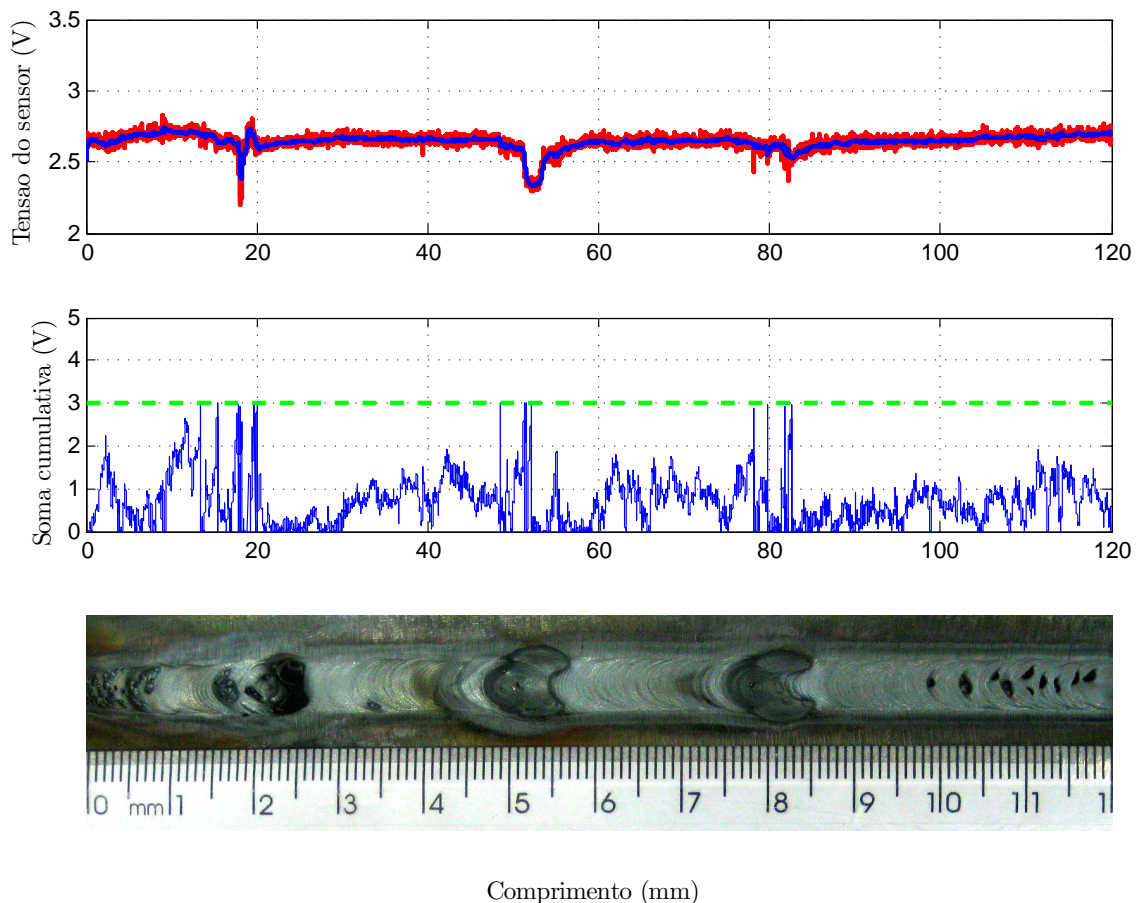
A figura 4.23 apresenta o resultado de um teste feito com um *stand-off* de 3 mm e com descontinuidades, no caso furos, alocadas relacionadas ao início da soldagem, aproximadamente entre: i) 19-28 mm, ii) 52-59 mm e iii) 80-90 mm.



**Figura 4.23.** Resultado do ajuste para a detecção de descontinuidades por furos. *stand-off* = 3 mm e descontinuidades em: 19-28 mm, 52-59 mm e 80-90 mm.

Na experiência da figura 4.23 o 100% das descontinuidades foram identificadas, gerando muitos alarmes dentro da região da descontinuidade.

A figura 4.24 apresenta o resultado de um teste feito com um *stand-off* de 5 mm e com descontinuidades alocadas, relacionadas ao início da soldagem, aproximadamente em: i) 21-27 mm, ii) 50-58 mm e iii) 78-86 mm.



**Figura 4.24.** Resultado do ajuste para a detecção de descontinuidades por furos. *stand-off* = 5 mm e descontinuidades em: 21-27 mm, 50-58 mm e 78-86 mm.

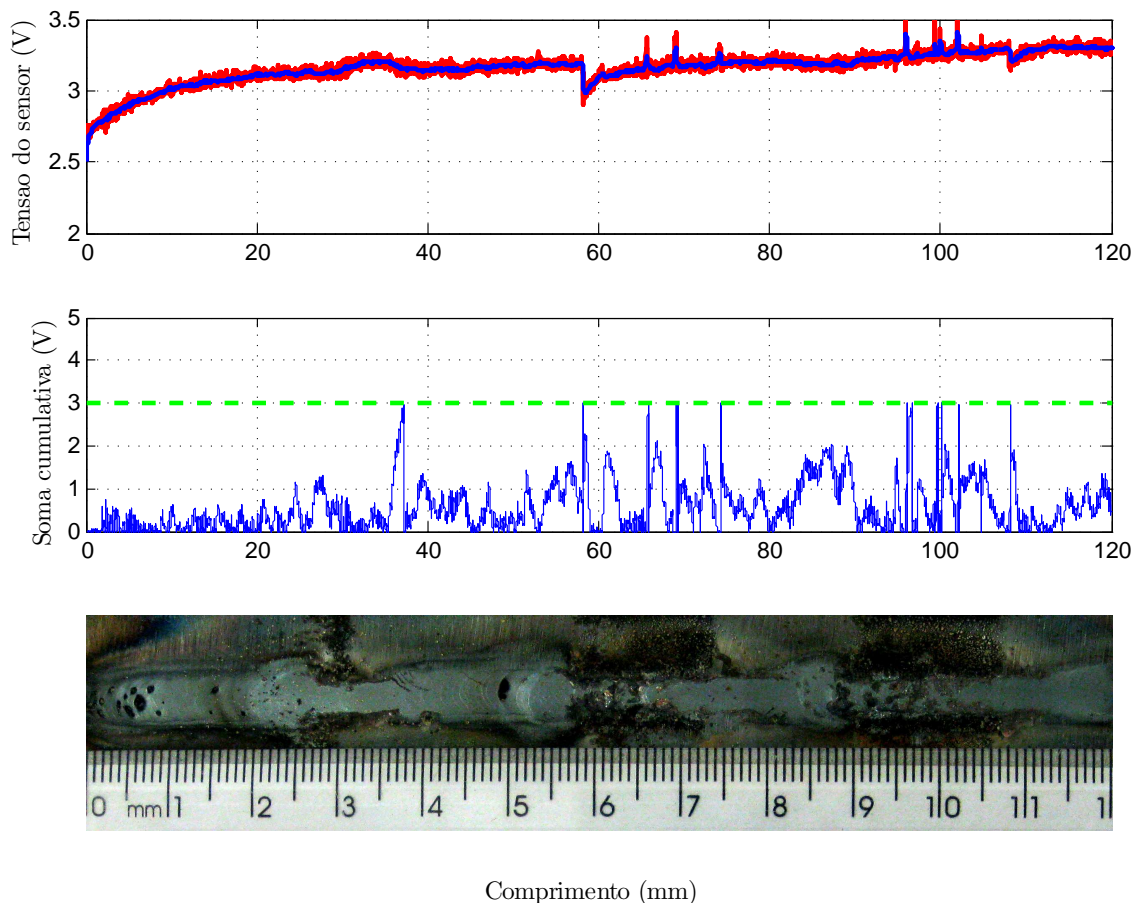
O resultado desta experiência é igual à anterior com este mesmo tipo de descontinuidade, os alarmes foram gerados durante a presença da descontinuidade.

As figuras 4.23 e 4.24 apresentam para a soma cumulativa uma resposta mais ativa na região das descontinuidades, gerando desta maneira todos os alarmes nas descontinuidades, tanto para *stand-off* de 3 e 5 mm. Nos cordões de solda destas experiências apresentaram pequenos furos não planejados que são atribuídos a erros aleatórios da fonte de soldagem no fornecimento do gás, para os que a soma cumulativa não atinge valores da mesma magnitude que para os furos fazendo que não sejam considerados no ajuste.

### Descontinuidades por presença de areia

A figura 4.25 apresenta o resultado de um teste feito com um *stand-off* de 3 mm e com descontinuidades alocadas relacionadas ao início da soldagem aproximadamente entre: i) 21-30 mm, ii) 60-70 mm e iii) 90-105 mm.





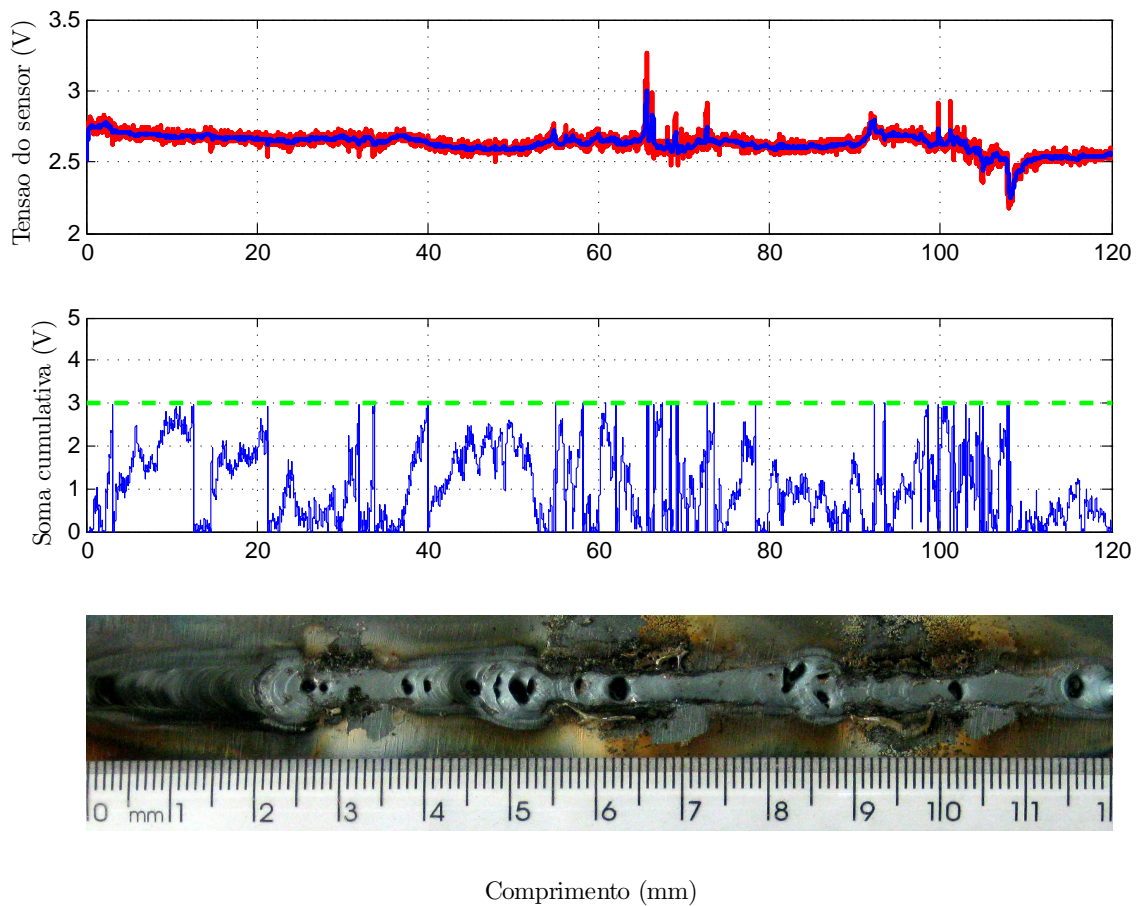
**Figura 4.25.** Resultado do ajuste para a detecção de descontinuidades por presença de areia. *stand-off* = 3 mm e descontinuidades em: 21-30 mm, 60-70 mm e 90-105 mm.

Duas das três descontinuidades são detetadas, a 'ii)' e a 'iii)', para a 'ii)' durante toda elas geraram alarmes, enquanto que para 'iii)' justo depois que iniciou-se a descontinuidade os alarmes foram gerados. A descontinuidade 'i)' não é detetada com sucesso devido a quantidade de areia presente em ela, embora gera um cordão defeituoso e quando este se iniciou com melhor qualidade (37 mm) um alarme foi gerada.

A figura 4.26 apresenta o resultado de um teste feito com um *stand-off* de 5 mm e com descontinuidades alocadas relacionadas ao início da soldagem aproximadamente entre: i) 22-35 mm, ii) 55-70 mm e iii) 90-105 mm.

Excetuando as descontinuidades detetadas até 20 mm (alarmes falsos), o sistema consegue detetar as descontinuidades gerando alarmes durante sua duração. Também este é o pior dos cordões, contribuído pela presença de descontinuidades não programadas (pequenos furos no cordão) geradas pela fonte de soldagem no fornecimento do gás.





**Figura 4.26.** Resultado do ajuste para a detecção de discontinuidades por presença de areia. *stand-off* = 5 mm e discontinuidades em: 22-35 mm, 55-70 mm e 90-105 mm.

Comparando os resultados apresentados nas figuras 4.25 e 4.26, pode-se ver que a discontinuidade por areia é a mais contaminante dos três tipos de discontinuidades apresentadas neste documento, portanto o que gera os cordões mais defeituosos.

Ajustado o algoritmo para a detecção de mudanças para o senso favorável para cada tipo de discontinuidade apresentada, seus parâmetros ( $R$ ,  $Q$ ,  $\nu$  e  $h$ ) são introduzidos no código vhdl para serem embarcados no FPGA e fazer os testes de detecção *on-line* de discontinuidades no processo de soldagem, os resultados obtidos estão em seguida.

### 4.2.3 Validação do sistema de detecção de descontinuidades

A parte final dos resultados encontra-se a validação do sistema de detecção de descontinuidades no processo de soldagem GTAW. Relembrando que os resultados das descontinuidades detectadas (quantidade e distância) foram fornecidos pelo FPGA, e que o computador só acompanhou o processo. O tipo de descontinuidades se mantém (arame, furos e areia) igual que os parâmetros de soldagem presentes na tabela 3.3.

Antes de continuar é relevante esclarecer que, durante a validação do sistema de detecção de descontinuidades surgiram alguns problemas na detecção das descontinuidades, consistente em um incremento na geração de alarmes falsos e atribuídos a problemas de instrumentação na bancada de soldagem, o que impediu realizar a validação para todos os casos de interesse da maneira como foi descrita na seção 3.2, foi possível realizar apenas um teste para descontinuidades por arame e com um *stand-off* de 5 mm. Para os demais tipos de descontinuidades usou-se o método descrito na seção 3.1.5 pela figura 3.3, ou seja, foram simulados os cordões de solda. As sugestões para solucionar o problema na bancada é apresentado na seção de conclusões e futuros trabalhos.

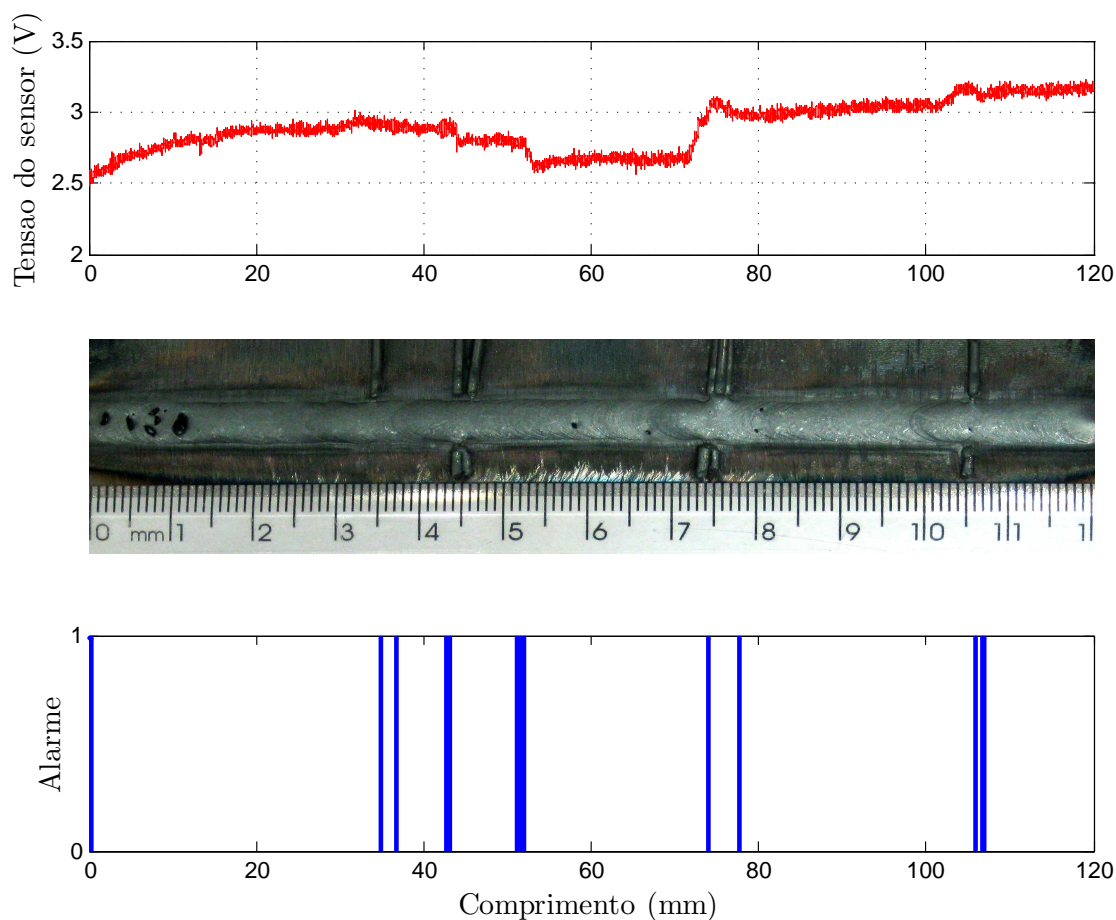
Neste contexto, os dados usados para simular o sensor infravermelho foram os mesmos apresentados previamente no ajuste do sistema de detecção de mudança, excepto para o caso da descontinuidade por arame com um *stand-off* de 5 mm

Os resultados para validação estão expostos conforme o tipo de descontinuidade. Para cada descontinuidade apresentam-se duas figuras, uma para cada distância (3 e 5 mm). Cada figura contém na parte (a) o sinal do sensor infravermelho que foi lido pelo computador, na parte (b) contém a fotografia do cordão e na parte (c) possui a informação fornecida pelo FPGA indicando o local de onde os alarmes por descontinuidade se originaram.

É importante destacar que para as descontinuidades introduzidas pelos arames, não necessariamente serão detectadas os locais alocados inicialmente. Isto é devido ao material introduzido ser fundido e pode-se deslocar por ação da possa de soldagem e gerar alguma protuberância.

#### Descontinuidades por arames (*on-line*)

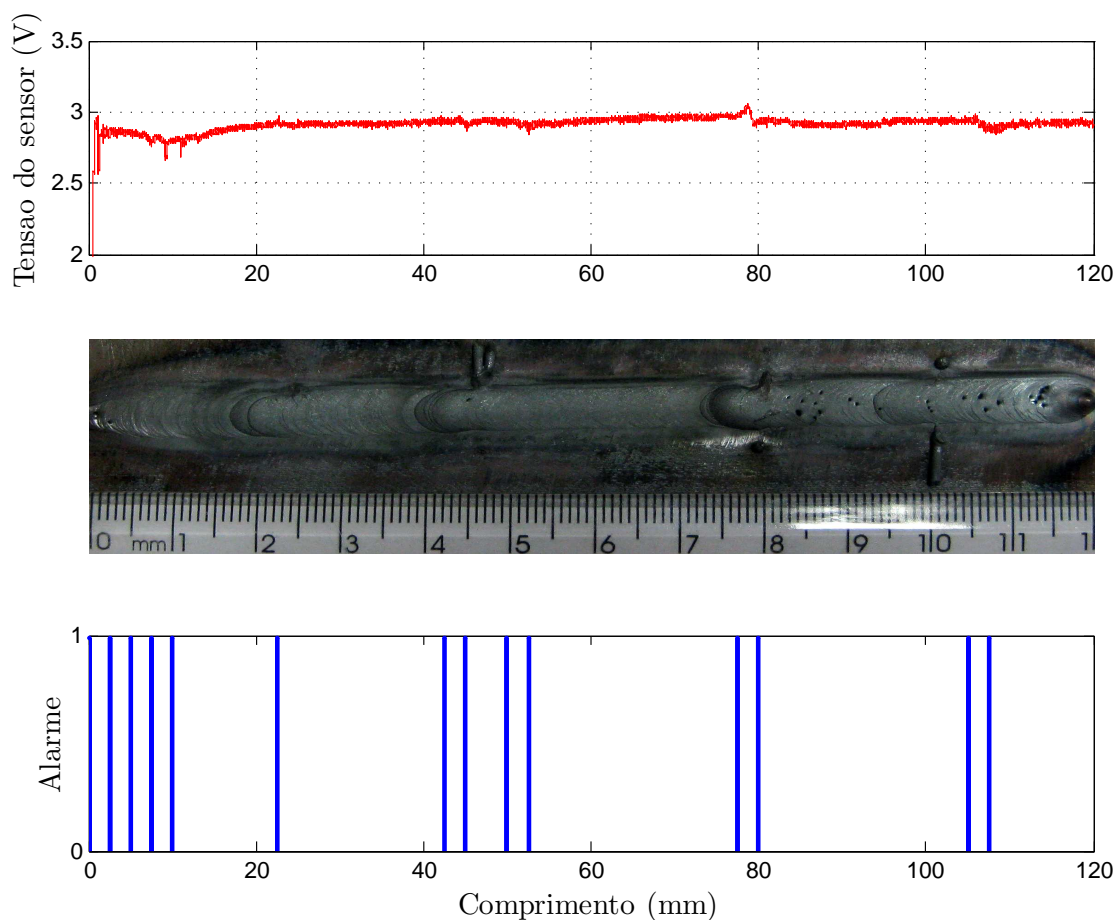
A figura 4.27 apresenta o resultado de um teste de validação para um *stand-off* de 3 mm. As descontinuidades foram alocadas relacionadas ao início da soldagem, aproximadamente nos seguintes par distância/quantidade-de-arames: i) 35 mm/1, ii) 45 mm/2, iii) 75 mm/3 e iv) 105mm/1.



**Figura 4.27.** Resultado da detecção *on-line* de discontinuidades geradas por arame. *stand-off* = 3 mm e discontinuidades no par distância/quantidade-de-arama: 35 mm/1, 45 mm/2, 75 mm/3 e 105mm/1.

Na figura 4.27 pode-se observar que os alarmes foram gerados perto ou na posição onde as discontinuidades foram alocadas, pode-se dizer que todas as discontinuidades foram detetadas pelo sistema de detecção de discontinuidades, mesmo que fossem gerados alarmes aproximadamente em 52 mm.

A figura 4.28 apresenta o resultado de um teste de validação para um *stand-off* de 5 mm. As discontinuidades foram alocadas, relacionadas ao início da soldagem, aproximadamente nos seguintes par distância/quantidade-de-arama: i) 25 mm/1, ii) 45 mm/2, iii) 80 mm/3 e iv) 100 mm/1.

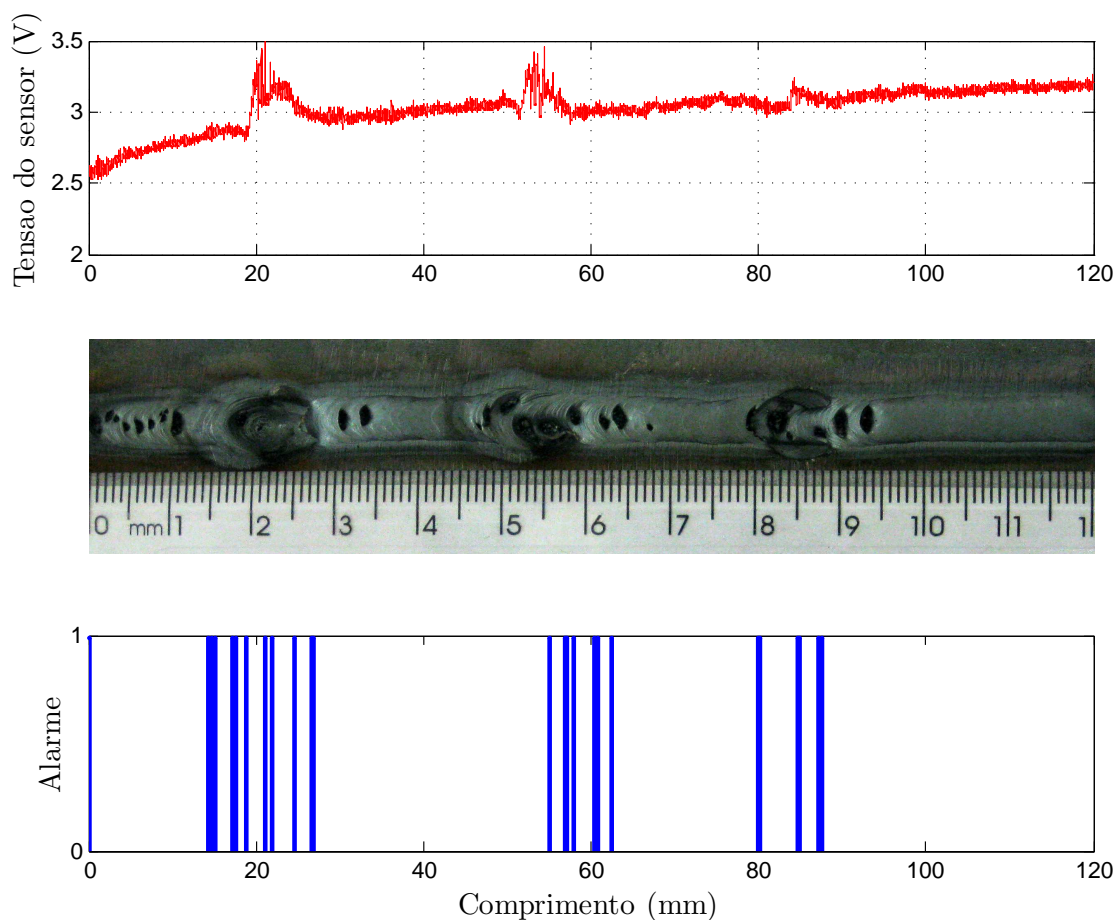


**Figura 4.28.** Resultado da detecção *on-line* de descontinuidades geradas por arame. *stand-off* = 5 mm e descontinuidades no par distância/quantidade-de-aramé: 25 mm/1, 45 mm/2, 80 mm/3 e 100 mm/1.

Os primeiros alarmes nesta experiência ocorrem devido à diferença entre o valor inicial dado para o algoritmo de detecção e os primeiros dados de tensão do sensor infravermelho, além de que no início o cordão não é uniforme. Exceptuando a descontinuidade ‘iv)’ para a qual o alarme foi gerado alguns milímetros depois de onde a descontinuidade foi alocada, as demais descontinuidades geraram uma alarme certo.

### Descontinuidades por furos (*on-line*)

A figura 4.29 apresenta o resultado do teste de validação para um *stand-off* de 3 mm. As descontinuidades foram alocadas, relacionadas ao início da soldagem, aproximadamente entre: i) 19-28 mm, ii) 52-59 mm e iii) 80-90 mm.

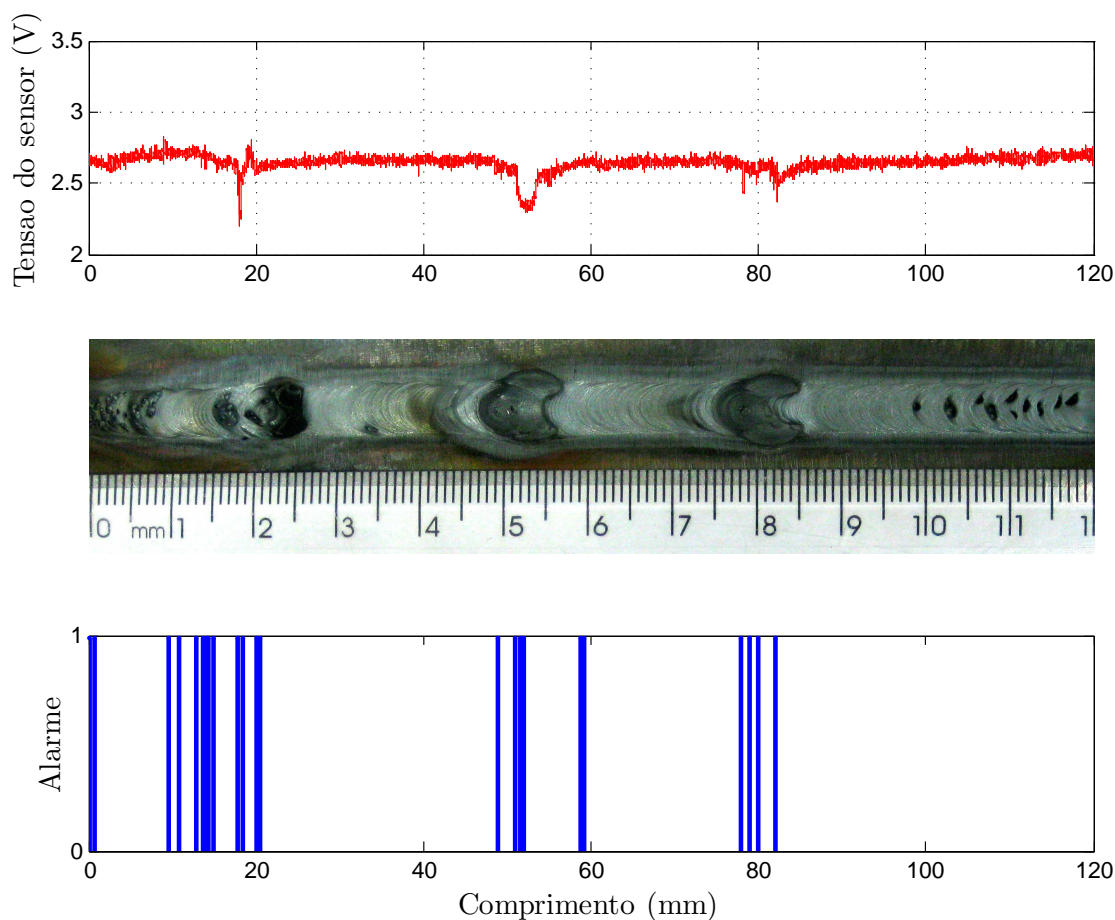


**Figura 4.29.** Resultado da detecção *on-line* de descontinuidades geradas por furos. *stand-off* = 3 mm e descontinuidades entre: 19-28 mm, 52-59 mm e 80-90 mm.

O 100% das descontinuidades foram identificadas, os alarmes gerados nesta experiência encontram-se em torno cada uma das descontinuidades.

A figura 4.30 apresenta o resultado de um teste de validação para um *stand-off* de 5 mm. As descontinuidades foram alocadas, relacionadas ao início da soldagem, aproximadamente entre: i) 21-27 mm, ii) 50-58 mm e iii) 78-86 mm.

Ao contrário que para o *stand-off* de 3 mm, para esta experiência apresentam-se alarmes falsos, mas as descontinuidades também geraram seus alarmes.

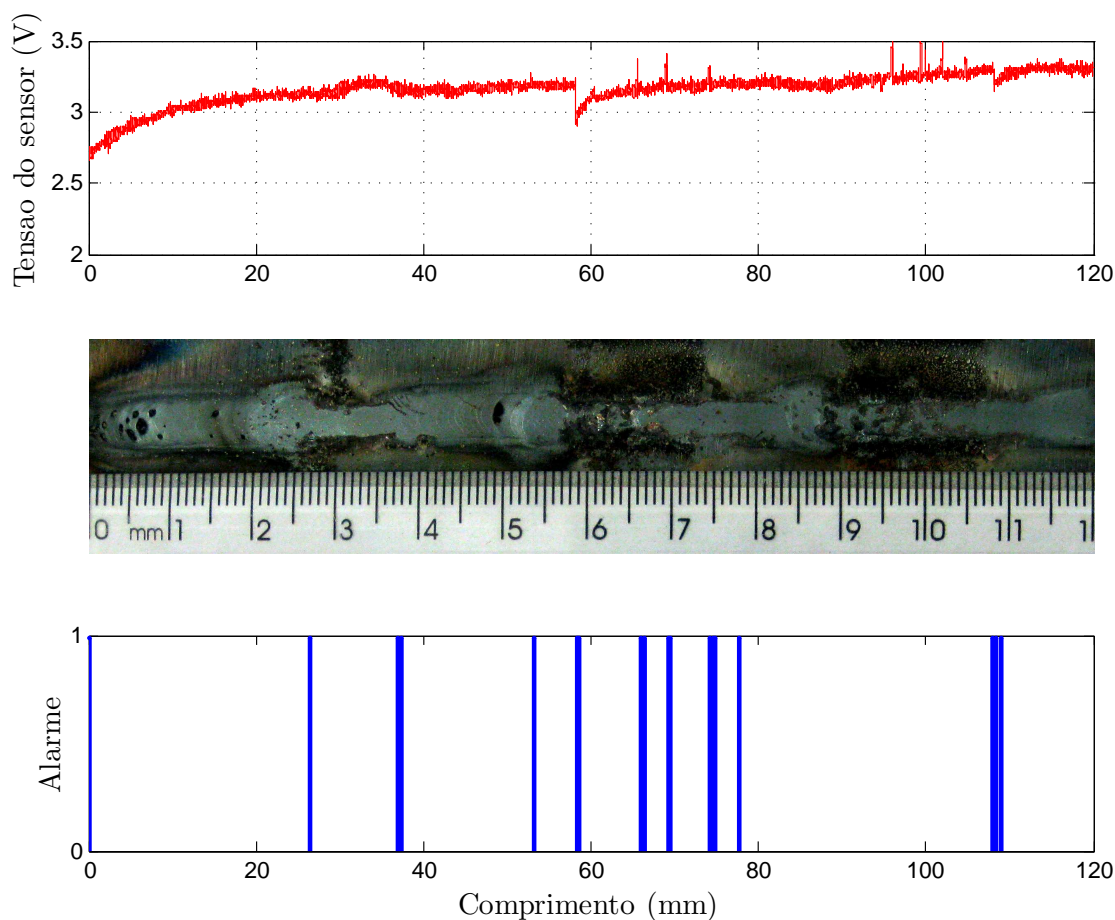


**Figura 4.30.** Resultado da detecção *on-line* de descontinuidades geradas por furos. *stand-off* = 5 mm e descontinuidades em: 21-27 mm, 50-58 mm e 78-86 mm.

### Descontinuidades por presença de areia (*on-line*)

A figura 4.31 apresenta o resultado de um teste de validação para um *stand-off* de 3 mm. As descontinuidades aconteceram, relacionadas ao início da soldagem, aproximadamente entre: i) 21-30 mm, ii) 60-70 mm e iii) 90-105 mm.

Para as descontinuidades 'i)' e 'ii)' foram gerados os alarmes enquanto estas ocorriam, mas para a descontinuidade 'iii)' o alarme é gerado logo depois que a descontinuidade por areia termina, mas tem-se uma variação no cordão que gera um alarme, que é o mesmo caso dos alarmes gerados aproximadamente em 38 mm.



**Figura 4.31.** Resultado da detecção *on-line* de descontinuidades geradas por presença de areia. *stand-off* = 3 mm e descontinuidades entre: 21-30 mm, 60-70 mm e 90-105 mm.

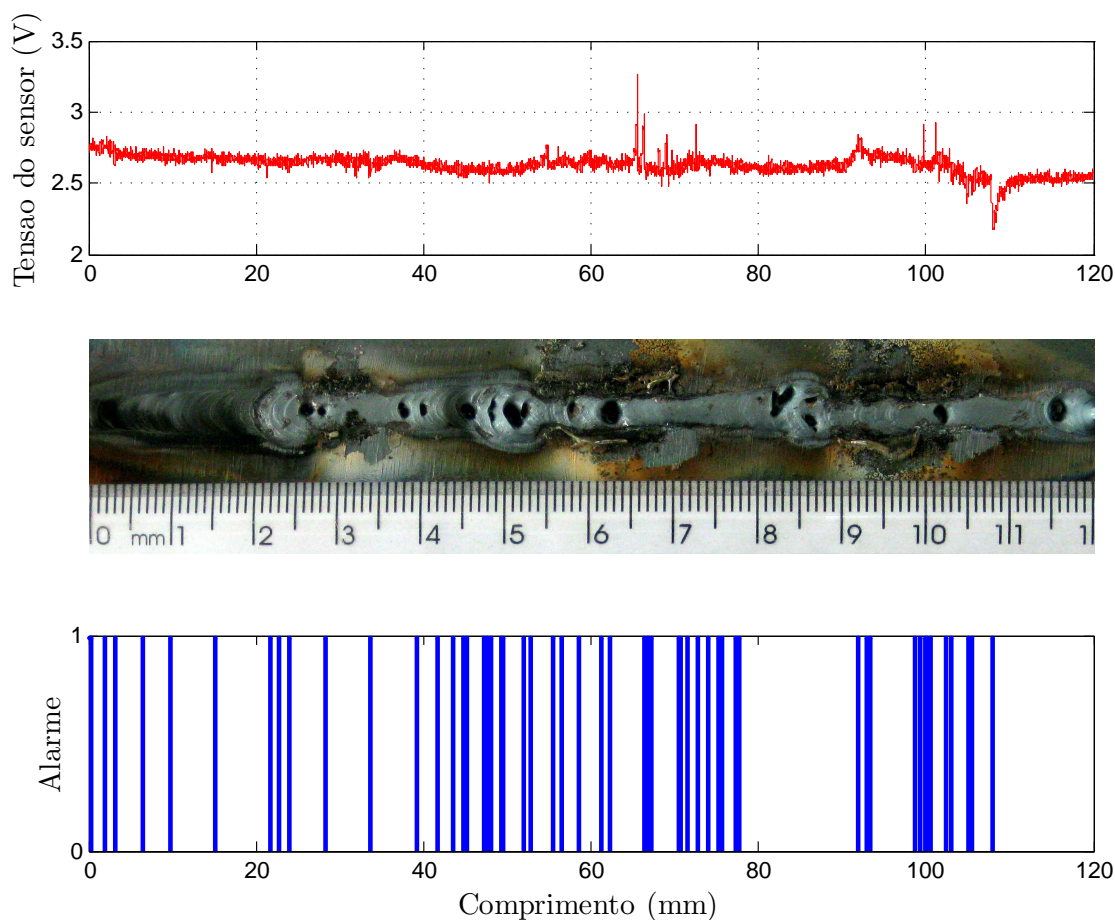
A figura 4.32 apresenta o resultado de um teste de validação para um *stand-off* de 5 mm. As descontinuidades aconteceram, relacionadas ao início da soldagem, aproximadamente entre: i) 22-35 mm, ii) 55-70 mm e iii) 90-105 mm.

Considerando os alarmes detetados nesta última experiência, este cordão tem que ser eliminado, a quantidade de alarmes gerados tanto por as descontinuidades como por os alarmes falsos é muito alta, a eliminação do cordão resulta também evidente pelo aspecto do cordão.

### 4.3 DISCUSSÃO DOS RESULTADOS

Neste trabalho se propôs um sistema embarcado em hardware usando arquiteturas reconfiguráveis baseadas em FPGAs (*Field Programmable Gate Array*) para a detecção *on-line* de descontinuidades no processo de Soldagem a Arco-Gás com eletrodo de Tungstênio (GTAW), mediante o sensoriamento óptico da radiação infravermelha do seu arco voltagem, usando um





**Figura 4.32.** Resultado da detecção *on-line* de descontinuidades geradas por presença de areia. *stand-off* = 5 mm e descontinuidades entre: 22-35 mm, 55-70 mm e 90-105 mm.

algoritmo de detecção de mudança composto por um filtro Kalman e um teste estatístico por somas cumulativas.

Para o desenvolvimento deste trabalho foi utilizada a técnica de co-projeto de Hardware/Software (*Hardware/Software codesign*), em que a parte hardware foi projetada para acelerar a execução dos periféricos no qual estão descritos os algoritmos de detecção de mudança. Já na parte software, foi implementado o processador *soft MicroBlaze* para o gerenciamento dos periféricos projetados e o sistema geral de detecção de descontinuidades.

O uso de um processador *soft* permitiu um ganho no tempo de desenvolvimento do projeto, possibilitando desta maneira focar-se no desenvolvimento dos algoritmos em hardware —no caso deste trabalho, o algoritmo de *detecção de mudança* (**Detector**), bem como nos periféricos para o *gerenciamento de tempo* (**gerTempo**), a *aquisição dos dados* (**conversaoAD**), e *memória das descontinuidades* (**memRAM**).

Neste trabalho foi implementada uma arquitetura *pipeline* de dois estágios para o desenvol-



vimento do algoritmo de detecção de mudança (**Detector**). Dita arquitetura permitiu verificar o uso dos FPGAs na detecção *on-line* de descontinuidades na área de soldagem.

O tempo total consumido pelo periférico **conversaoDetector** é  $4,23 \mu\text{s}$ , dos quais  $3,17 \mu\text{s}$  (73,38%), corresponde ao tempo consumido para fazer a conversão analógica/digital de dois valores de entrada e obter a sua média. Adicionalmente, o periférico **Detector** possui um tempo de execução de  $0,85 \mu\text{s}$  (19,68%). Note-se que o tempo de execução na aquisição e conversão é maior do que o tempo de execução no bloco de tratamento dos dados (**Detector**). Dito comportamento é típico dos sistemas de processamento de dados.

Os resultados de síntese demonstraram que a arquitetura proposta requer um grande número de multiplicadores dedicados no dispositivo FPGA. Isto pode ser explicado levando em conta as operações aritmética em 32 bit executadas (soma/subtração, multiplicação e divisão). Entretanto, os demais recursos de hardware utilizados (*slices*, *Flip-flops* e *LUTs*) apresentam um consumo de aproximadamente metade dos recursos totais disponíveis.

Nas experiências feitas na bancada de soldagem se observou que para o *stand-off* de 5 mm foi apresentada uma quantidade maior de alarmes falsos do que para o *stand-off* de 3 mm, devido ao sinal do sensor infravermelho ter apresentado maior instabilidade para o *stand-off* de 5 mm.

Dentre as descontinuidades estudadas neste trabalho, o teste de descontinuidades de furos apresentou uma melhor resposta na detecção. Isto poderia ser atribuído hipoteticamente ao fato de que as demais descontinuidades utilizam outros materiais (arame e areia, respectivamente). Por outro lado, as descontinuidades por areia gerou mais alarmes falsos, sendo portanto a pior resposta de detecção observada. Contudo, o estudo sobre as diferenças entre as descontinuidades não está no escopo deste trabalho, sendo portanto uma possibilidade de futuras pesquisas neste tema.

Para as descontinuidades por arame percebeu-se que a localização e a quantidade de material utilizado por descontinuidade têm uma relação direta com a perturbação do sinal infravermelho utilizado.

Observou-se um deslocamento da localização dos alarmes das descontinuidades entre as experiências *off-line* (etapa de ajuste do algoritmo) e *on-line* (etapa de validação). Dito deslocamento é explicado pelo fato de ter sido utilizada a placa USB na parte *on-line*, cujo conversor digital/analógico gera o valor pré-estabelecido  $\pm 50 \text{ mV}$ , o que conseqüentemente gera uma variação na soma cumulativa.

## 5 CONCLUSÕES E FUTUROS TRABALHOS

### 5.1 CONCLUSÕES

- Mostrou-se que o sistema desenvolvido em hardware, utilizando arquiteturas reconfiguráveis baseadas em FPGAs, conseguiu detectar de maneira *on-line* descontinuidades produzidas no processo de soldagem GTAW.
- A técnica de co-projeto Hardware/Software permitiu dar um melhor enfoque ao problema de detectar descontinuidades no processo de soldagem GTAW gerando um ganho no desenvolvimento do trabalho aqui apresentado.
- Mostrou-se que o uso de um processador *soft* foi uma boa opção para o gerenciamento de periféricos projetados para a detecção de descontinuidades.
- Mostrou-se que com o algoritmo de detecção de mudança se aproveitam as características de paralelismo do FPGA.
- Para uma aplicação específica, em que se conheça a faixa os valores de entrada do sensor, pode-se mudar para uma aritmética de ponto fixo, diminuindo de esta maneira o consumo de recursos do FPGA.
- As melhores respostas para a detecção de descontinuidades se conseguiram para o *stand-off* de 3 mm.
- As descontinuidades melhor detectadas foram as produzidas por a presença de furos e as que pior resposta na detecção apresentaram foram as produzidas por a presença de areia.
- Notou-se que as descontinuidades por arame foram detectadas segundo a sua posição e a quantidade de arame presente no cordão.
- Ter desenvolvido um sistema para simular os cordões foi uma boa estratégia de depuração, permitiu validar o hardware e o algoritmo de detecção de mudança antes de levar para a bancada de soldagem.

- O sistema para simular os cordões também permite validar outros tipos de técnicas para a detecção *on-line* de descontinuidades na soldagem.

## 5.2 FUTUROS TRABALHOS

Para acrescentar o sistema desenvolvido para a detecção de descontinuidades no processo de soldagem GTAW e aproveitar as capacidades que neste campo oferecem os FPGAs, uma série de trabalhos poderão ser realizados no futuro.

Uma vez que para este trabalho foi utilizado somente um sensor para monitorar o processo de soldagem, e considerando as capacidades de paralelismo do FPGA, pode-se considerar novas pesquisas com adição de um ou mais sensores ao sistema, bem como com a utilização de outro tipo de algoritmo na monitoração.

Após o monitoramento, o passo seguinte que pode ser dado é o controle. A partir da informação gerada pelo sistema de detecção, ou seja, os alarmes, poderiam-se fazer ajustes na fonte de soldagem com o intuito de diminuir o efeito da descontinuidade na solda.

Conforme exposto anteriormente, a implementação do algoritmo de detecção de mudança usou periféricos com aritmética de 32 bit para soma/subtração, multiplicação e divisão, podendo ser utilizada para propósitos gerais. Em outras palavras, independentemente dos dados de entrada, o periférico pode ser usado, pois contém a descrição do algoritmo de detecção de mudança. Porém, em casos onde se queira uma aplicação específica, na qual seja conhecido o intervalo dos dados de entrada, pode ser realizado um estudo utilizando uma aritmética de ponto fixo, conseguindo assim uma diminuição em relação ao consumo de recursos do FPGA.

Para corrigir os erros que ocasionaram alarmes falsos nos testes de validação, se propõe reprojeter a bancada em que o hardware para a aquisição dos dados se encontra, fato que são muitas as modificações que em ele tem sido feitas com a variedade de projetos que no laboratório se tem, em outras palavras padronizar a bancada para aquisição dos sinais nos diferentes tipos de processos de soldagem estudados no laboratório.

## REFERÊNCIAS BIBLIOGRÁFICAS

- [1] FRANCO, F. D. *Monitorização e Localização de Defeitos na Soldagem TIG através do Sensoriamento Infravermelho*. Dissertação (Mestrado) — Universidade de Brasília. Faculdade de Tecnologia. Departamento de Engenharia Mecânica., 2008.
- [2] MIRAPEIX, J. et al. Real-time arc welding defect detection technique by means of plasma spectrum optical analysis. *NDT & E International*, v. 39, n. 5, p. 356 – 360, 2006. ISSN 0963-8695.
- [3] SFORZA, P.; BLASIIS, D. de. On-line optical monitoring system for arc welding. *NDT & E International*, v. 35, n. 1, p. 37 – 43, 2002. ISSN 0963-8695.
- [4] HARTENSTEIN, R.; KAISERSLAUTERN, T. Basics of reconfigurable computing. In: SPRINGER (Ed.). *Designing Embedded Processors*. Dordrecht, The Netherlands: Jörg Henkel and Sri Parameswaran, 2007. cap. 20, p. 451–501.
- [5] AWS, A. W. S. (Ed.). *Welding Handbook (Welding Technology)*. 8. ed. 1989.
- [6] NORRISH, J. *Advanced Welding Processes*. IOP Publishing, 1992.
- [7] MARQUES, P. V. *Tecnologia da soldagem*. ESAB S/A INDÚSTRIA E COMÉRCIO, 1991. 352 p.
- [8] DAVID, S. A.; BABU, S. S.; VITEK, J. M. Welding. In: BUSCHOW, K. H. J. et al. (Ed.). *Encyclopedia of Materials: Science and Technology*. Oxford: Elsevier, 2003. p. 1 – 9. ISBN 978-0-08-043152-9.
- [9] BRACARENSE, A. Q. *Processo de Soldagem TIG - GTAW*. Maio 2000.
- [10] MODENESI, P. J.; MARQUES, P. V. *Soldagem I, Introdução a os Processos de Soldagem*. Novembro 2000.
- [11] MODENESI, P. J. *Introdução à Física do Arco Elétrico e sua Aplicação na Soldagem dos Metais*. Fevereiro 2007.
- [12] HECHT, E. *Optics*. 4. ed. Addison Wesley, 2002.

- [13] FEYNMAN, R. P. *The Feynman Lectures on Physics*. 6. ed. Addison Wesley, 1977.
- [14] EISBERG, R. M. *Fundamentals of Modern Physics*. 3. ed. John Wiley & Sons, Inc, 1963.
- [15] ZHANG, Y. (Ed.). *Real-Time Weld Process Monitoring*. Woodhead Publishing Limited e CRC Press LLC, 2008.
- [16] FRADEN, J. Infrared thermometers. In: WEBSTER, J. G. (Ed.). *The Measurement, Instrumentation and Sensors Handbook*. CRC Press, 1999. cap. 32.6.
- [17] MICHALSKI, L. et al. *Temperature measurement*. 2. ed. John Wiley & Sons, Inc, 2001. 518 p.
- [18] GREWAL, M. S.; ANDREWS, A. P. *Kalman Filtering: Theory and Practice Using Matlab*. John Wiley & Sons, Inc, 2001.
- [19] GUSTAFSSON, F. *Adaptative Filtering and Change Detection*. John Wiley & Sons, Ltd, 2000. 510 p.
- [20] SIMON, D. *Optimal State Estimation*. John Wiley & Sons, Inc, 2006. 552 p.
- [21] WELCH, G.; BISHOP, G. *An Introduction to the Kalman Filter*. Chapel Hill, NC 27599-3175, 2001.
- [22] MEYER-BAESE, U. *Digital Signal Processing with FPGA*. Springer-Verlag, 2001.
- [23] CHU, P. P. *FPGA Prototyping by VHDL Examples*. John Wiley & Sons, Inc, 2008.
- [24] MAXFIELD, C. *FPGAs. World Class Designs*. Newnes, 2009.
- [25] DUBEY, R. *Introduction to Embedded System Design Using Field Programmable Gate Arrays*. Springer-Verlag, 2009.
- [26] COFER, R.; HARDING, B. F. *Rapid System Prototyping with FPGAs*. Burlington: Newnes, 2006. (Embedded technology). ISBN 978-0-75-067866-7.
- [27] XILINX. *MicroBlaze Processor Reference Guide*. Ug081 (v9.0). 2008.
- [28] VAHID, F.; STITT, G. Hardware/software partitioning. In: DEHON, S. H. A. (Ed.). *Reconfigurable Computing: The Theory and Practice of FPGA-Based Computation*. Morgan Kaufmann, 2008. cap. 26, p. 539–560.
- [29] IEEE. *IEEE Standard for Floating-Point Arithmetic*. 08 2008.
- [30] SÁNCHEZ, D. F. et al. Parameterizable floating-point library for arithmetic operations in fpgas. In: *SBCCI '09: Proceedings of the 22nd Annual Symposium on Integrated Circuits and System Design*. New York, NY, USA: ACM, 2009. p. 253–258. ISBN 978-1-60558-705-9.

- [31] HURTADO, R. H.; ALFARO, S. C. A.; LLANOS, C. H. A methodology for “on-line” monitoring system in a welding process using fpgas. In: *Industrial Technology (ICIT), 2010 IEEE International Conference on.* [s.n.], 2010. p. 162 –167.
- [32] XILINX. *Spartan-3 Generation FPGA User Guide.* Ug331 (v1.6). Dezembro 3 2009.
- [33] XILINX. *MicroBlaze Development Kit Spartan-3E 1600E Edition User Guide.* Ug257 (v1.1). December 5 2007.
- [34] NATIONAL INSTRUMENTS. *User Guide and Specifications USB-6008/6009.* July 2005.
- [35] DIGILENT. *Digilent Pmod AD1 Analog to Digital Module Converter Board Reference Manual.* 215 E Main Suite D Pullman, WA 99163, April 2005.
- [36] CALEX ELECTRONICS LTD. *TL-S, TL-GA – User Manual.* 2004.
- [37] ABB. *Technical catalogue 2009 - Chapter "Analog signal converters CC range".* 2009.
- [38] XILINX. *Spartan-3 Generation FPGA User Guide.* Ug331(v1.6). December 3 2009.

# Livros Grátis

( <http://www.livrosgratis.com.br> )

Milhares de Livros para Download:

[Baixar livros de Administração](#)

[Baixar livros de Agronomia](#)

[Baixar livros de Arquitetura](#)

[Baixar livros de Artes](#)

[Baixar livros de Astronomia](#)

[Baixar livros de Biologia Geral](#)

[Baixar livros de Ciência da Computação](#)

[Baixar livros de Ciência da Informação](#)

[Baixar livros de Ciência Política](#)

[Baixar livros de Ciências da Saúde](#)

[Baixar livros de Comunicação](#)

[Baixar livros do Conselho Nacional de Educação - CNE](#)

[Baixar livros de Defesa civil](#)

[Baixar livros de Direito](#)

[Baixar livros de Direitos humanos](#)

[Baixar livros de Economia](#)

[Baixar livros de Economia Doméstica](#)

[Baixar livros de Educação](#)

[Baixar livros de Educação - Trânsito](#)

[Baixar livros de Educação Física](#)

[Baixar livros de Engenharia Aeroespacial](#)

[Baixar livros de Farmácia](#)

[Baixar livros de Filosofia](#)

[Baixar livros de Física](#)

[Baixar livros de Geociências](#)

[Baixar livros de Geografia](#)

[Baixar livros de História](#)

[Baixar livros de Línguas](#)

[Baixar livros de Literatura](#)  
[Baixar livros de Literatura de Cordel](#)  
[Baixar livros de Literatura Infantil](#)  
[Baixar livros de Matemática](#)  
[Baixar livros de Medicina](#)  
[Baixar livros de Medicina Veterinária](#)  
[Baixar livros de Meio Ambiente](#)  
[Baixar livros de Meteorologia](#)  
[Baixar Monografias e TCC](#)  
[Baixar livros Multidisciplinar](#)  
[Baixar livros de Música](#)  
[Baixar livros de Psicologia](#)  
[Baixar livros de Química](#)  
[Baixar livros de Saúde Coletiva](#)  
[Baixar livros de Serviço Social](#)  
[Baixar livros de Sociologia](#)  
[Baixar livros de Teologia](#)  
[Baixar livros de Trabalho](#)  
[Baixar livros de Turismo](#)