



Universidade Federal de Campina Grande

Centro de Engenharia Elétrica e Informática

Curso de Graduação em Engenharia Elétrica

ANDRÉA DE ANDRADE MENDES

RELATÓRIO DE ESTÁGIO SUPERVISIONADO

Campina Grande, Paraíba
Maio de 2016

ANDRÉA DE ANDRADE MENDES

RELATÓRIO DE ESTÁGIO SUPERVISIONADO

*Relatório de Estágio Supervisionado submetido à
Unidade Acadêmica de Engenharia Elétrica da
Universidade Federal de Campina Grande como
parte dos requisitos necessários para a obtenção do
grau de Bacharel em Ciências no Domínio da
Engenharia Elétrica.*

Área de Concentração: Controle e Automação

Orientador:

Professor George Acioli Júnior, D. Sc.

Campina Grande, Paraíba
Maio de 2016

ANDRÉA DE ANDRADE MENDES

RELATÓRIO DE ESTÁGIO SUPERVISIONADO

Relatório de Estágio Supervisionado submetido à Unidade Acadêmica de Engenharia Elétrica da Universidade Federal de Campina Grande como parte dos requisitos necessários para a obtenção do grau de Bacharel em Ciências no Domínio da Engenharia Elétrica.

Área de Concentração: Controle e Automação

Aprovado em ____ / ____ / _____

Universidade Federal de Campina Grande
Avaliador

Professor George Acioli Júnior, D. Sc.
Universidade Federal de Campina Grande
Orientador, UFCG

Dedico este trabalho aos meus pais, aos meus irmãos e aos meus sobrinhos.

AGRADECIMENTOS

Agradeço a toda minha família, em especial aos meus pais, Edivânia e José Ribeiro, e meus irmãos, Leandro e Júnior, pelo apoio incondicional. E também aos meus sobrinhos, Elias Henrique e João Lucas, por alegrar o meu dia sempre que possível.

Aos professores do Departamento de Engenharia Elétrica, por todo conhecimento repassado, em especial aos professores George Acioli Júnior, por ter aceitado me orientar durante este trabalho e Péricles Rezede Barros, pelo acompanhamento durante o estágio.

A todos os colegas de laboratório que me auxiliaram sempre que precisei.

À minha noiva e companheira de todas as horas, Carol, por seu amor e dedicação, e a todos meus amigos que me acompanharam e me deram força para enfrentar essa jornada.

A todos vocês, meu muito obrigado.

RESUMO

Este relatório apresenta as atividades da aluna Andréa de Andrade Mendes realizadas durante o Estágio Supervisionado no Laboratório de Instrumentação Eletrônica e Controle (LIEC), do Departamento de Engenharia Elétrica (DEE), da Universidade Federal de Campina Grande (UFCG), sob orientação do professor George Acioli Junior e supervisão do professor Péricles Rezende de Barros. As atividades apresentadas neste relatório referem-se ao estudo do equipamento Controlador Lógico Programável (CLP) do fabricante HI Tecnologia, dos softwares SPDSW e MATLAB, e à elaboração de documento para descrição do funcionamento da esteira.

Palavras-chave: CLP, Ladder, MATLAB, SPDSW, Simulink.

LISTA DE FIGURAS

Figura 1- LIEC.	12
Figura 2 - CLP da família ZAP91X.	13
Figura 3 - Janela inicial do SPDSW.	14
Figura 4 - Esteira Transportadora.	15
Figura 5 - Motor DC.	16
Figura 6 - Trecho de Programa Ladder para atribuir valores a S000.	17
Figura 7 - Encoder.	18
Figura 8 - Sensor óptico de barreira.	18
Figura 9 - Entradas correspondentes aos sensores de barreira.	19
Figura 10 - Módulo de aquecimento.	19
Figura 11 - Leitura da temperatura do módulo de aquecimento.	20
Figura 12 – Trecho de código em ladder para atribuir valores máximo e mínimo ao <i>dutycycle</i> do PWM às resistências do módulo de aquecimento.	20
Figura 13 – Gráfico que relaciona o <i>dutycycle</i> mínimo à frequência programada.	21
Figura 14 – Gráfico relacionando valores de P3 do bloco FQG e o valor percentual da tensão.	21
Figura 15 - Módulo de Resfriamento.	22
Figura 16 – Trecho de código em ladder referente à leitura de temperatura e acionamento dos coolers do módulo de resfriamento.	23
Figura 17 – Janela inicial do Programa HS1 Power Tool.	27
Figura 18 – Janela do programa HS1Power Tool para configuração do canal de comunicação.	28
Figura 19 - Janela do programa HS1Power Tool para configuração dos parâmetros equipamento para efetuar a comunicação.	29
Figura 20 - Janela do programa HS1Power Tool para configuração dos parâmetros equipamento para efetuar a comunicação.	30
Figura 21 – Bloco OPC Configuration.	31
Figura 22 – Janela para configuração do bloco OPC configuration.	32
Figura 23 – Bloco OPC Read.	32
Figura 24 - Janela para configuração do bloco OPC Read.	33

Figura 25 - Bloco OPC Write.....	34
Figura 26 - Janela para configuração do bloco OPC Write.....	34
Figura 27 – Programa no Simulink para escrita e leitura de variáveis para identificação do sistema do módulo de aquecimento.....	35
Figura 28 - Gráfico da temperatura do módulo de aquecimento (°C) para diversos degraus aplicados ao dutycycle do PWM da resistência em função do tempo.....	36
Figura 29 - Janela inicial do System Identification.....	37
Figura 30 - Janela para ajuste dos parâmetros da Função de Transferência.....	38
Figura 31 - Gráfico com todas as funções de transferência geradas.....	39
Figura 32 - Equação da FT5.....	40
Figura 33 – Controlador PID gerado pela toolbox PID Tuner para a função de transferência para temperatura do módulo de aquecimento.....	41

LISTA DE TABELAS

Tabela 1 - Faixa de valores que determinam o sentido da rotação do motor DC.....	17
Tabela 2 - Entradas e saídas	23
Tabela 3 - Variáveis auxiliares	24
Tabela 4 - Constantes	25
Tabela 5 - Fitting das Funções de Transferência para diversos valores de pólos e zeros.	38

SUMÁRIO

1	Introdução.....	11
2	Local do Estágio.....	11
3	Atividades Realizadas.....	12
3.1	CLP (Controlador Lógico Programável).....	13
3.1.1	CLP da Família ZTK ZAP91X	13
3.1.2	Software SPDSW	14
3.1.3	Programação na Liguagem Ladder.....	14
3.2	Visão Geral da Esteira transportadora.....	15
3.2.1	Motor DC	16
3.2.2	Encoder.....	17
3.2.3	Sensores Ópticos de Barreira	18
3.2.4	Módulo de Aquecimento	19
3.2.5	Módulo de Resfriamento	21
3.3	Descrição Das Entradas e Saídas.....	23
3.4	HS1 Power Tool	26
3.5	Comunicação MATLAB/CLP.....	30
3.6	Identificação do Sistema de Temperatura	36
3.7	PID Tuning	40
3.8	Monitoria de Sistema de Automação Industrial	41
4	Conclusão.....	41
4.1	Sugestões para Trabalhos Futuros.....	41
	REFERÊNCIAS.....	422

1 INTRODUÇÃO

O objetivo deste relatório é apresentar as atividades desenvolvidas, durante o estágio supervisionado, pela aluna Andréa de Andrade Mendes no Laboratório de Instrumentação Eletrônica e Controle (LIEC), situado na Universidade Federal de Campina Grande (UFCG). O período de vigência do estágio supervisionado foi de 15/02/2016 até 13/05/2016 totalizando 210h (7 créditos), sendo a carga horária de 18 horas semanais, sob a orientação do professor George Acioli Júnior e supervisão do professor Péricles Rezende Barros.

2 LOCAL DO ESTÁGIO

Laboratório de Instrumentação Eletrônica e Controle – LIEC, Figura 1. Este é um laboratório do Departamento de Engenharia Elétrica cujos integrantes trabalham nas áreas de Controle, Automação e Eletrônica. Os projetos de pesquisa contam com a participação dos alunos do curso e do corpo docente. Dessa forma, o embasamento teórico e prático necessário ao futuro Engenheiro Eletricista é desenvolvido e solidificado. O LIEC está situado no Setor C da Universidade Federal de Campina Grande em Campina Grande - Paraíba. Possui uma área de aproximadamente 600 m² com oito Laboratórios de desenvolvimento, duas salas de apoio técnico, uma moderna sala de apresentação de trabalhos, salas para Pós-graduação e professores. Modernas instalações e um grupo de pessoas altamente qualificado fazem do LIEC um ambiente bastante agradável para se trabalhar e pesquisar. [1]



Figura 1- LIEC.
Fonte: [1]

3 ATIVIDADES REALIZADAS

As atividades de estágio foram baseadas no Trabalho de Conclusão de Curso do Engenheiro Ezequiel de Souza Batista [2], que teve como objetivo a concepção de uma esteira transportadora. Algumas atividades foram desenvolvidas de forma paralela, mas pode-se enumerá-las da seguinte forma:

1. Estudo do programa em Ladder, desenvolvido por Ezequiel de Souza Batista, carregado no CLP;
2. Entendimento das funcionalidades gerais da esteira transportadora;
3. Desenvolvimento de documento explicando de forma detalhada as entradas e saídas do sistema da esteira transportadora;
4. Comunicação do CLP com o Simulink para leitura e escrita de variáveis utilizando a *toolbox* OPC do MATLAB;
5. Identificação do sistema do módulo de aquecimento utilizando a *toolbox System Identification* do MATLAB;
6. Desenvolvimento de controlador PID para temperatura do módulo de aquecimento utilizando a *toolbox PID Tuner* do MATLAB;

7. Acompanhamento dos alunos da disciplina Sistemas de Automação Industrial com auxílio do professor orientador.

3.1 CLP (CONTROLADOR LÓGICO PROGRAMÁVEL)

A definição segundo a norma IEC 11311-1 afirma que o CLP é um sistema eletrônico digital, desenvolvido para uso em ambiente industrial, que usa memória programável para armazenamento interno de instruções do usuário. Controladores Lógicos Programáveis, também chamados de Controladores Programáveis ou PLCs, pertencem à família dos computadores de estado sólido, usando circuitos integrados, em vez de dispositivos eletromecânicos, para implementar funções de controle. Eles são capazes de armazenar instruções tais como: sequenciamento, temporização, contagem, aritmética, manipulação de dados e comunicação, para controlar máquinas e processos industriais [3].

3.1.1 CLP DA FAMÍLIA ZTK ZAP91X

O modelo utilizado na aplicação pertence ao kit de treinamento CLP - ZTK ZAP91X do fabricante HI Tecnologia (Figura 2).



Figura 2 - CLP da família ZAP91X.

Fonte: [4]

Os Kits de Treinamento em programação de CLP's baseados nos controladores ZAP91X possuem interface homem máquina integrada, interface de sinais para entrada e saídas do tipo digital e analógico e modelo com canais seriais e/ou ethernet. [4]

A documentação disponibilizada pela empresa facilita a utilização do equipamento.

3.1.2 SOFTWARE SPDSW

O software utilizado para carregar os programas no CLP foi o SPDSW versão 3.3.03, Figura 3, disponibilizado gratuitamente pela HI Tecnologia.

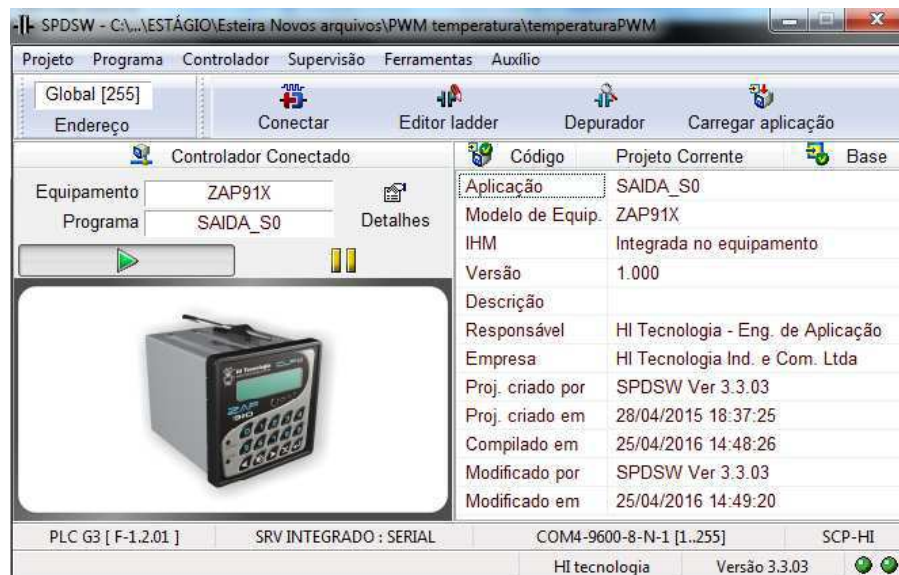


Figura 3 - Janela inicial do SPDSW.

Não foi necessário instalar o software, pois a máquina utilizada já estava configurada, inclusive a comunicação e a conexão das entradas e saídas.

3.1.3 PROGRAMAÇÃO NA LIGUAGEM LADDER

Neste tópico serão listadas as principais funções utilizadas neste trabalho na programação do CLP, a linguagem de programação é o LADDER, o entendimento destas funções é de extrema importância para seu correto funcionamento.

- MOV
- ADD
- SUB
- MUL
- DIV
- FQF
- FCT
- BLQ
- TMR
- AND

As funções abordadas aqui são pertencentes ao software SPDSW e compatíveis com os controladores da HI Tecnologia. A documentação disponibilizada pelo fabricante [5] e o *help* do software SPDSW fornecem informações mais detalhes sobre cada bloco.

3.2 VISÃO GERAL DA ESTEIRA TRANSPORTADORA

O objeto de estudo foi uma esteira transportadora de tamanho reduzido construída pelo aluno Ezequiel com finalidade de desenvolver seu Trabalho de Conclusão de Curso.

A estrutura física da esteira transportadora foi reaproveitada de uma esteira da marca CALOI modelo FITNESS. [2]

Na figura 4, identifica-se os principais elementos que compõem a esteira, que são: 1 - motor DC; 2 - encoder 3, 4, 5 e 6 - sensores ópticos de barreira; 7 - câmara de resfriamento; 8 - câmara de aquecimento e 9 - fonte de alimentação.

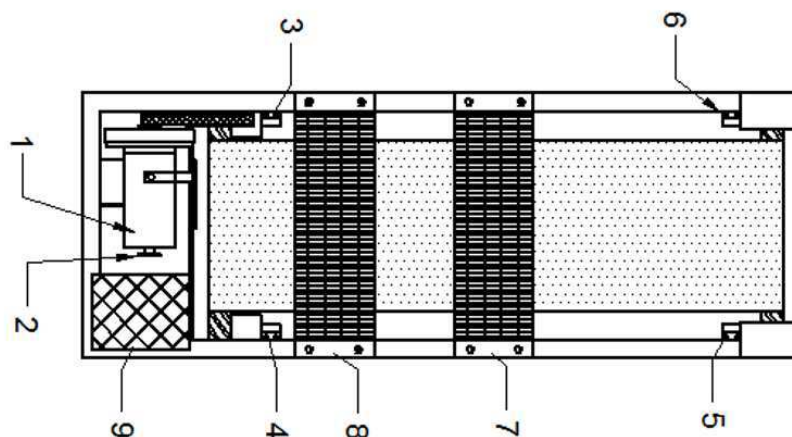


Figura 4 - Esteira Transportadora.
Fonte: [2]

As entradas e saídas da esteira estão conectadas a um CLP e podem ser controladas e/ou medidas.

3.2.1 MOTOR DC

O modelo do motor (Figura 5) é o original da esteira CALOI e possui as seguintes especificações:

- Modelo: MOTOR PMDC;
- Tensão: 180 V DC (podendo operar com tensões inferiores);
- Potência: 1HP;
- Fabricante: Universal Motors (Indústria Brasileira)



Figura 5 - Motor DC.

No programa em Ladder, Figura 6, o motor DC foi acionado para mover-se no sentido horário ou anti-horário, com diferentes velocidades, seguindo os valores da saída analógica S0000, com *tag* MIV-S-301, entre 0 e 4095, o qual fornece um valor de tensão (0 - 10V).

Quando a chave I0003 é ligada, atribui-se o valor de D0000 a S0000 e quando está desligada, atribui-se o valor 1650, que faz com que ele pare.

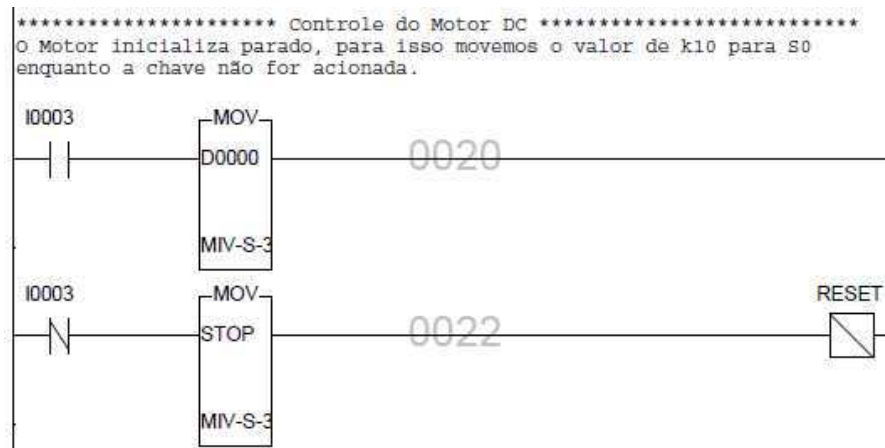


Figura 6 - Trecho de Programa Ladder para atribuir valores a S000.

De acordo com a referência [2], pode-se determinar as faixas para diferentes valores de S0000, correspondentes aos sentidos horário e anti-horário e a faixa que a esteira permanece parada. A partir disso, obteve-se a Tabela 1.

Tabela 1 - Faixa de valores que determinam o sentido da rotação do motor DC.

Estado	Faixa
Move-se no sentido horário	0-1250
Parado	1251-2740
Move-se no sentido anti-horário	2740-4095

3.2.2 ENCODER

A esteira possui um encoder incremental, Figura 7, para determinar a velocidade do Motor CC. O modelo do encoder corresponde ao "ww - 12 encoder" produzido pela empresa NU-BOTICS.



Figura 7 - Encoder.
Fonte: [6]

3.2.3 SENSORES ÓPTICOS DE BARREIRA

No sensor óptico de barreira que a esteira possui, Figura 8, foi utilizado como elemento emissor um LED alto brilho azul e, como elemento receptor, um foto transistor TIL81 com circuito dedicado.



Figura 8 - Sensor óptico de barreira.
Fonte: [2]

Os sensores de barreira foram conectados às entradas digitais I0008 a I0011 e acionam LEDs (saídas digitais O0004 a O0007) do CLP, como pode ser visto no trecho de código em linguagem Ladder da Figura 9.

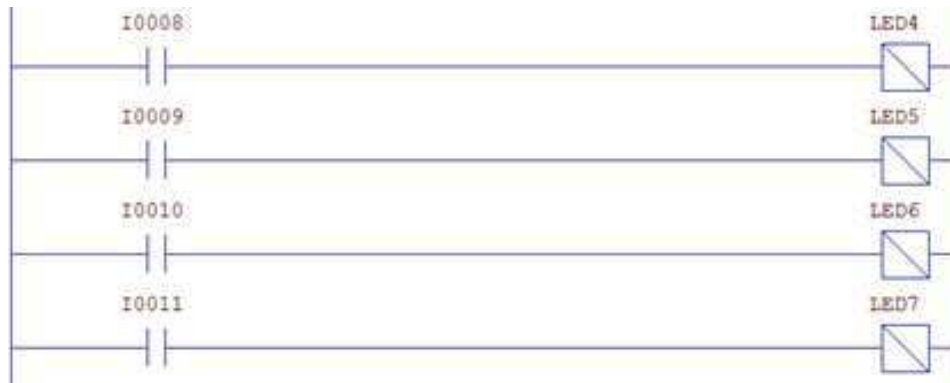


Figura 9 - Entradas correspondentes aos sensores de barreira.

3.2.4 MÓDULO DE AQUECIMENTO

Como pode-se ver na Figura 10, o módulo de aquecimento possui duas resistências de 1800 W e dois *coolers*, responsáveis pela variação térmica no interior da câmara, e um sensor de temperatura LM35, para realizar a medição da temperatura. Além disso, existe um sensor de barreira, o par emissor e receptor.

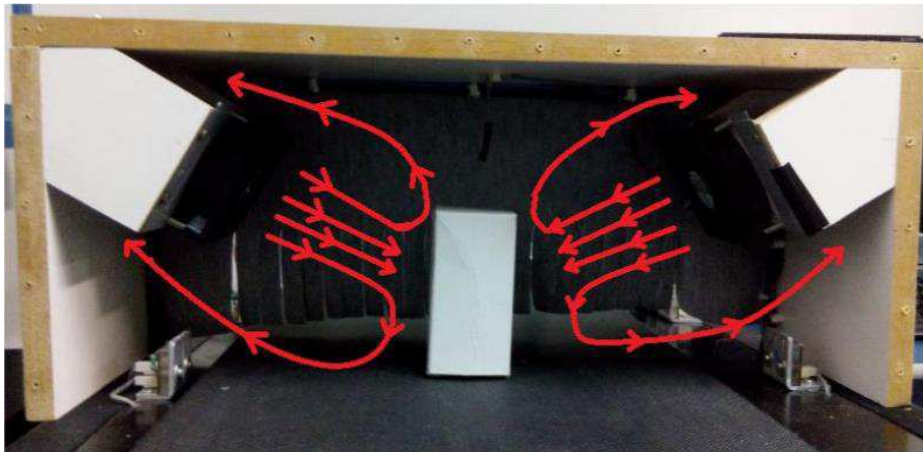


Figura 10 - Módulo de aquecimento.

Fonte: [2]

Pode-se identificar no código carregado no CLP, Figura 11, que a *tag* TT20 refere-se à leitura de tensão do sensor de temperatura enviada pela entrada digital E0006. Esse valor varia entre 0 mV e +10 mV e necessita de conversão (1) para unidade de engenharia (°C).

$$T_{\circ C} = \frac{10 * T_{lido}}{409,5} \quad (1)$$

Após a conversão, o valor é armazenado em outra variável, D0015, cuja *tag* é TQ CELSIUS.

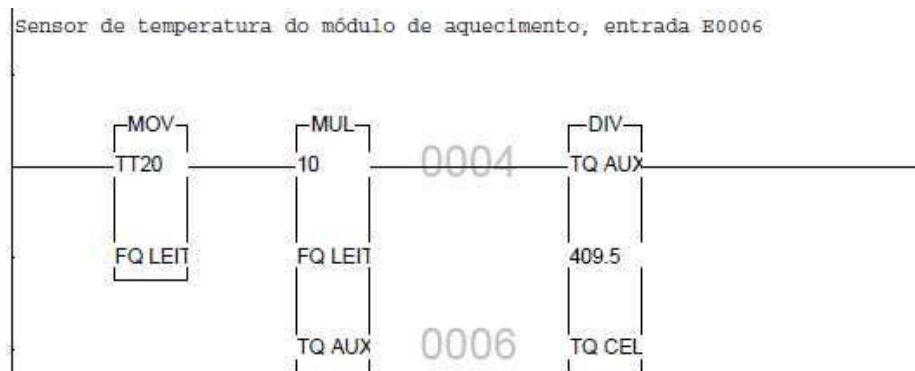


Figura 11 - Leitura da temperatura do módulo de aquecimento.

Para teste, utilizou-se a chave I0002 para atribuir dois valores, DUTYMAX (104) e DUTYMIN (7), correspondentes aos ciclos de trabalho de 100% e de 0%, respectivamente. Por isso, disponibilizou-se, para esse trecho de código, Figura 12, a possibilidade de atribuir os valores máximo ou mínimo ao *DutyCycle*.

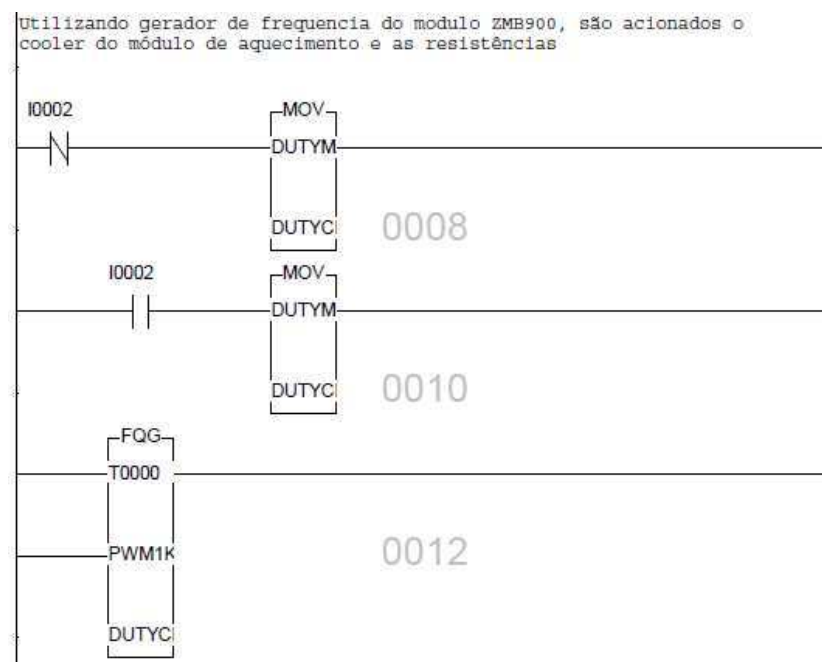


Figura 12 – Trecho de código em ladder para atribuir valores máximo e mínimo ao *dutycycle* do PWM às resistências do módulo de aquecimento.

O PWM foi construído para uma frequência de 1 kHz. Como pode ser conferido pela Figura 13, o valor mínimo do duty cycle (P3) deve ser de 7 no bloco FQG. [7]

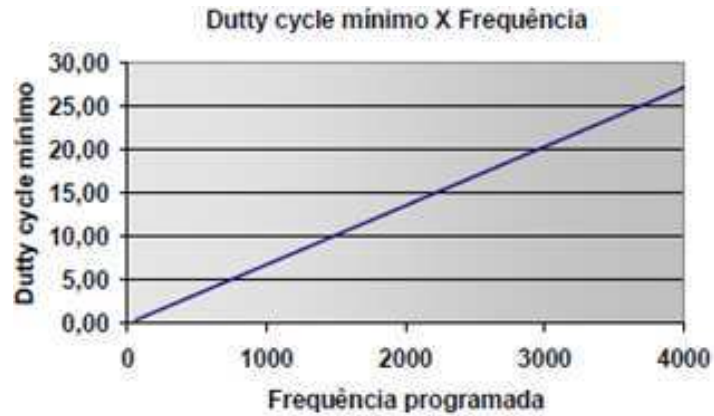


Figura 13 – Gráfico que relaciona o *dutty cycle* mínimo à frequência programada.
Fonte: [7]

A partir de medições dos valores exatos de tensão, variando entre 0 V e 24 V, sendo 24 V equivalente à 100%, para correspondentes a valores de *duty cycle* maiores que 7, obteve-se o gráfico da figura 14.

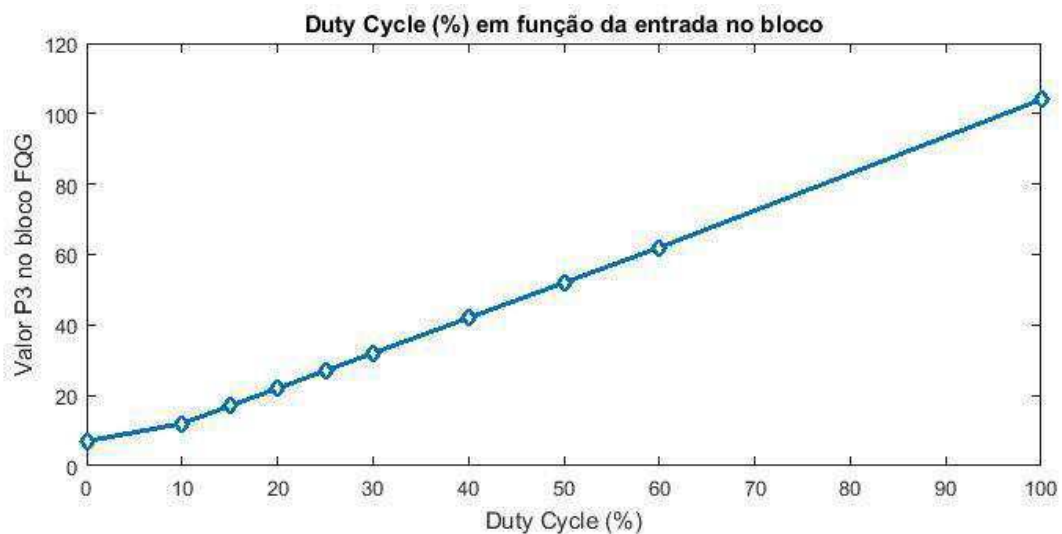


Figura 14 – Gráfico relacionando valores de P3 do bloco FQG e o valor percentual da tensão.

Quando a FQG está sendo usada, o LED da saída digital O0000 é acionado. Os *coolers* são acionados quando o PWM recebe qualquer valor maior que 7.

3.2.5 MÓDULO DE RESFRIAMENTO

O módulo de resfriamento, Figura 15, é composto por um sensor de temperatura LM35 e dois *coolers*. A leitura da temperatura do sensor e o acionamento dos *coolers* são realizados utilizando uma entrada analógica e uma saída digital, respectivamente.



Figura 15 - Módulo de Resfriamento.
Fonte [2]

A leitura da temperatura do módulo de resfriamento é análoga à do módulo de aquecimento, inclusive a conversão de unidade (1).

O sensor LM35 está conectado ao CLP pela entrada analógica E0007, *tag* TT10. Após a conversão o valor é armazenado na variável TF CELSIUS.

Os *coolers* são acionados simultaneamente e sem controle da velocidade de rotação, pela saída digital O0010, identificada pela *tag* VENTFRIO.

O trecho de código referente à leitura da temperatura e acionamento dos coolers do módulo de resfriamento encontra-se na Figura 16.

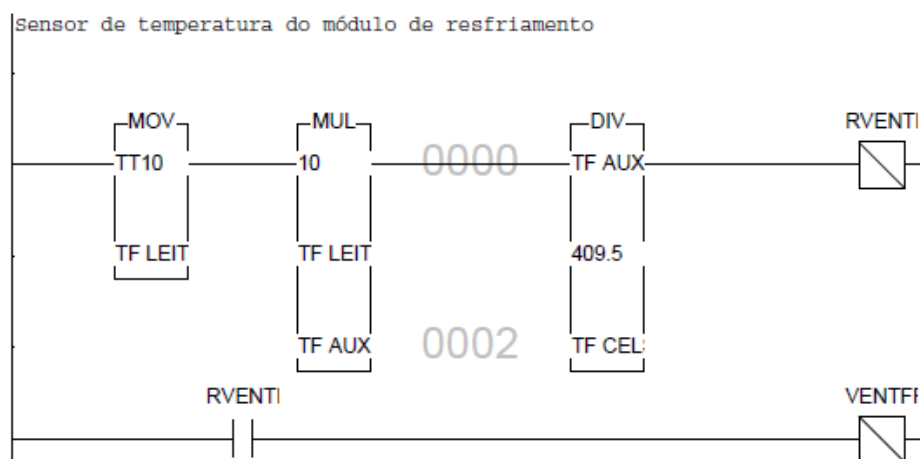


Figura 16 – Trecho de código em ladder referente à leitura de temperatura e acionamento dos coolers do módulo de resfriamento.

3.3 DESCRIÇÃO DAS ENTRADAS E SAÍDAS

Com a finalidade de facilitar o entendimento e visualização das variáveis do sistema, fez-se um conjunto de tabelas com todas as entradas, saídas e constantes, contendo descrição da variável, *tag* e endereço. A *tag* é uma maneira mais simples de atribuir um nome que signifique o que a variável representa.

As variáveis de entrada podem ser dos tipos analógica (E) ou digital (I). Da mesma forma são as variáveis de saída, analógica (S) ou digital (O).

As variáveis de entrada e de saída são as mais importantes e representam a comunicação entre a planta, sensores e atuadores, e o CLP.

Descreveu-se as variáveis de entrada e saída na Tabela 2.

Tabela 2 - Entradas e saídas

Descrição da variável	Tag	Endereço
Entradas Digitais		
On/Off do motor DC	-	I0003
Sensor óptico de barreira	-	I0008
Sensor óptico de barreira	-	I0009
Sensor óptico de barreira	-	I0010
Sensor óptico de barreira	-	I0011
Entradas Analógicas		
Sensor de temperatura do módulo de aquecimento	TT20	E0006
Sensor de temperatura do módulo de resfriamento	TT10	E0007
Saídas Digitais		
Saída digital que aciona cooler do módulo de resfriamento	VENTFRIO	O0010
LED acionado por sensor de barreira	LED4	O0004
LED acionado por sensor de barreira	LED5	O0005
LED acionado por sensor de	LED6	O0006

barreira		
LED acionado por sensor de barreira	LED7	O0007
Saída Analógica		
Saída Analógica para a placa do inversor que controla o motor	MIV-S-301	S0000

As variáveis auxiliares podem ser do tipo contato auxiliar (R), memória interna (M – 16 bits) ou memória real (D – 32 bits). Esses tipos de variáveis são responsáveis por operações de conversão de unidade e armazenamento na memória do CLP.

Encontram-se na Tabela 3 as variáveis auxiliares.

Tabela 3 - Variáveis auxiliares

Descrição da variável	Tag	Endereço
Contatos Auxiliares		
Aciona os coolers do módulo de resfriamento	RVENTFRIO	R0000
Reseta o FTC quando o motor estiver parado	RESET	R0001
Memória interna (16 bits)		
Número de voltas por transição, até completar o valor de 32767	MAXVOLTAS	M0004
Contador de pulsos canal 0	PULENCODER	M0005
PWM para velocidade do motor	M00	M0011
Duty cycle do PWM do bloco gerador de frequência (módulo ZMB900)	DUTY	M0015
Memória real (32 bits)		
Valor do conversor D/A para saída S0 (controla a velocidade do motor)		D0000
Variável que guarda valor da leitura analógica do sensor de temperatura do módulo	TF LEITURA	D0010
Variável auxiliar do módulo de resfriamento	TF AUX1	D0011
Temperatura do módulo de resfriamento em unidade de	TF CELSIUS	D0012

engenharia (°C)		
Variável que guarda valor da leitura analógica do sensor de temperatura do módulo	FQ LEITURA	D0013
Variável auxiliar do módulo de aquecimento	TQ AUX	D0014
Temperatura do módulo de aquecimento em unidade de engenharia (°C)	TQ CELSIUS	D0015

As constantes podem ser do tipo constante inteira (K) ou constante real (Q). As constantes são muito úteis no armazenamento de valores importantes, fixos, para conversão de unidades, ou que representam um estado. Por exemplo, da Tabela 4, tem-se o valor STOP, que é um valor importante, pois, quando atribuído à saída do conversor AD do motor DC, faz com que a esteira pare.

Tabela 4 - Constantes

Constantes Inteiras			
Descrição da variável	Tag	Valor	Endereço
Resolução do encoder	128	128	K0000
Gerador de frequência PWM	PWM1K	1000	K0001
Valor correspondente a zero do PWM	DUTYMIN	7	K0002
Valor correspondente a 100% do PWM	DUTYMAX	104	K0003
Valor máximo de memória inteira por divisão	32767	32767	K0004
Multiplica o valor VC de temperatura	10	10	K0006
Valor que para o motor	STOP	1650	K0010
Constantes Reais			
Distância percorrida a cada pulso	DISTPUL	0.709	Q0002
Valor de conversão para o cálculo de temperatura	409.5	409.5	Q0004

3.4 HS1 POWER TOOL

Para realizar a identificação do sistema do módulo de aquecimento e, finalmente, projetar um controlador, era necessário comunicar um software que permitisse atribuir diversos valores à variável de entrada e ler a temperatura.

Para isso, utilizou-se o Simulink do MATLAB e era necessário criar uma interface para conectá-los: MATLAB e CLP.

Para servir como interface para comunicação do MATLAB com o CLP, utilizou-se o driver OPC HS1, desenvolvido pela HI Tecnologia. Por meio dele, foi possível criar um cliente OPC para receber e enviar dados via comunicação serial. Para esta versão, os recursos disponibilizados são para escrita e leitura de variáveis do tipo R, M ou D. [8]

Os procedimentos necessários para instalação foram efetuados seguindo as recomendações da Nota de Aplicação ENA00033 da HI Tecnologia. Os passos relatados a seguir referem-se apenas à utilização do software.

Passo 1: Carregar a aplicação desejada no CLP utilizando a programa SPDSW.

Passo 2: Fechar o programa SPDSW ou outro que esteja se comunicando com o CLP pela serial.

Passo 3: Abrir o programa HS1 Power Tool e clicar em *Connect*, Figura 17.

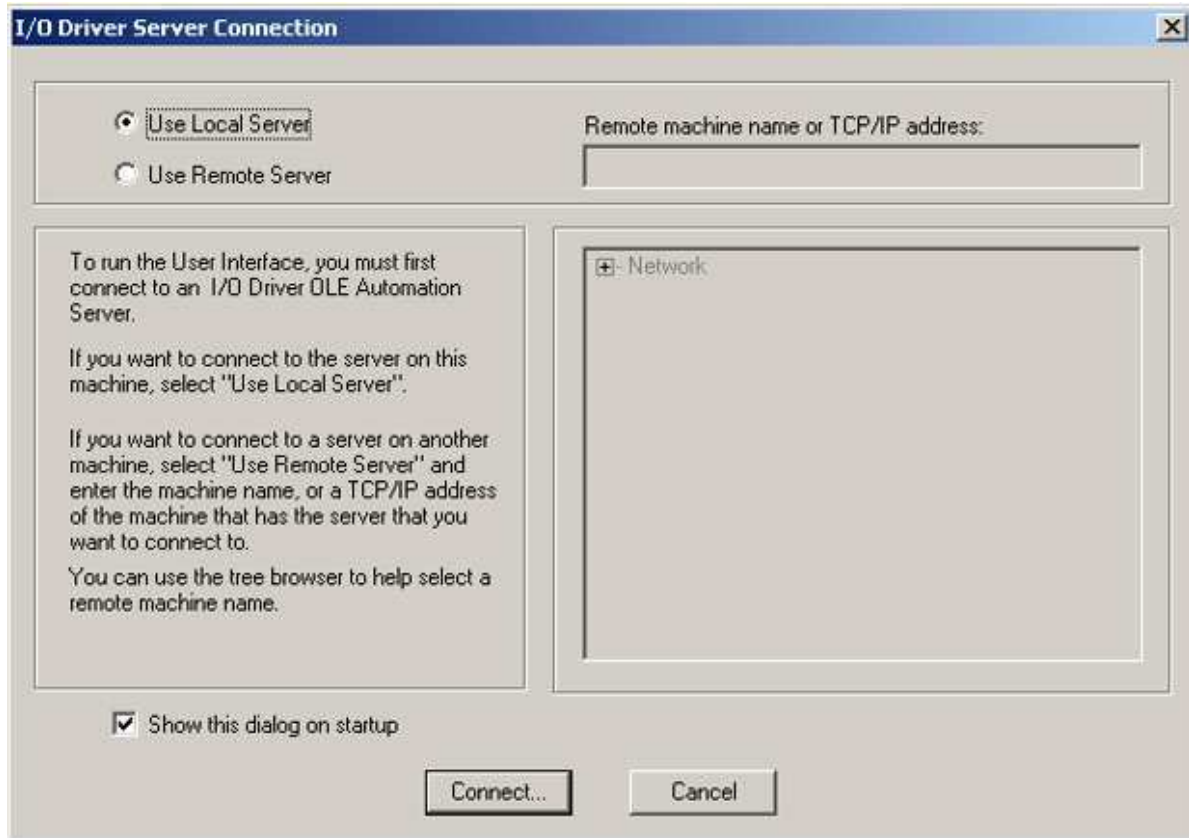


Figura 17 – Janela inicial do Programa HS1 Power Tool.
Fonte [8]

Passo 4: Criar novo Channel(), habilitá-lo e ajustar os parâmetros do canal, Figura 18.

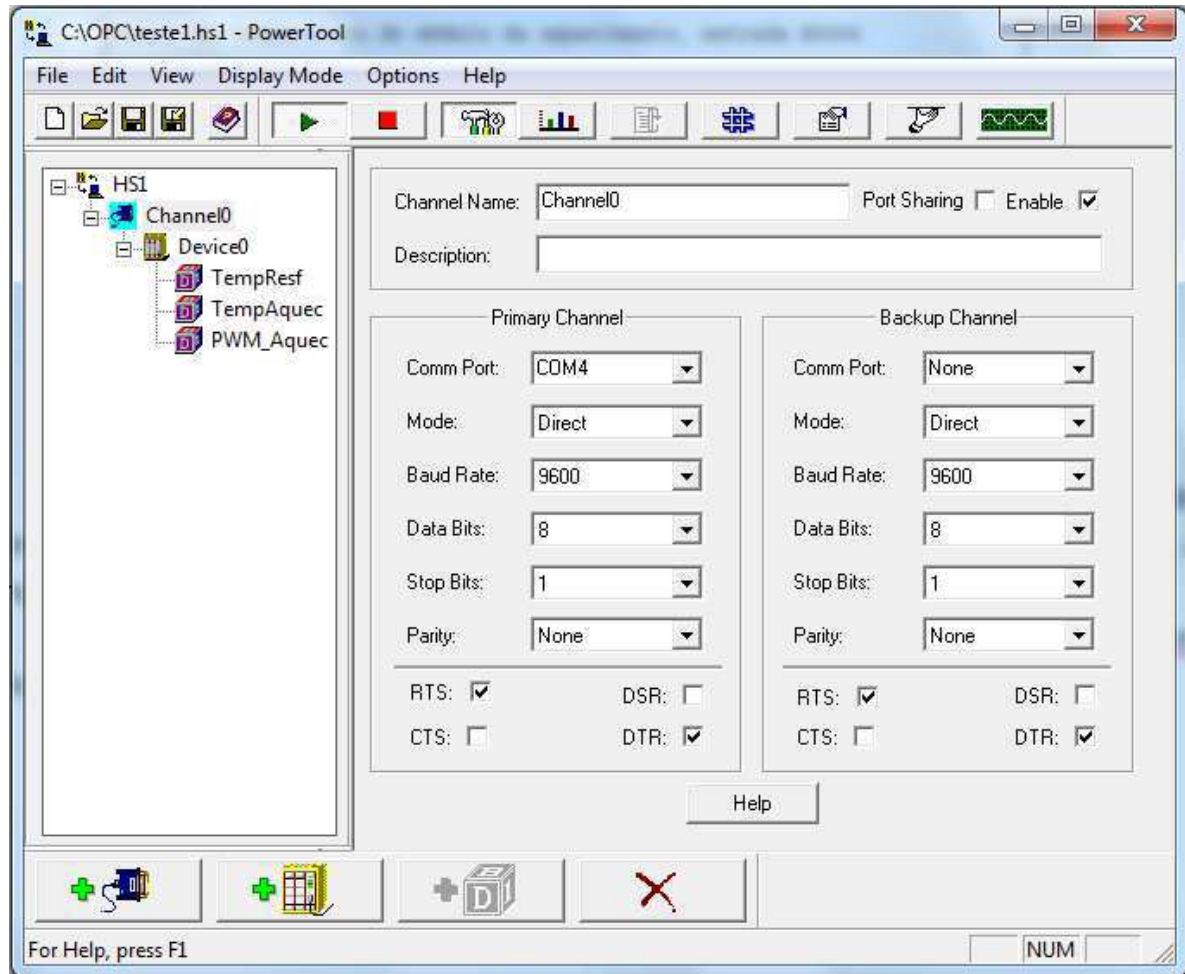


Figura 18 – Janela do programa HS1Power Tool para configuração do canal de comunicação.

Passo 4: Criar novo Device(), habilitá-lo e ajustar os parâmetros, Figura 19.

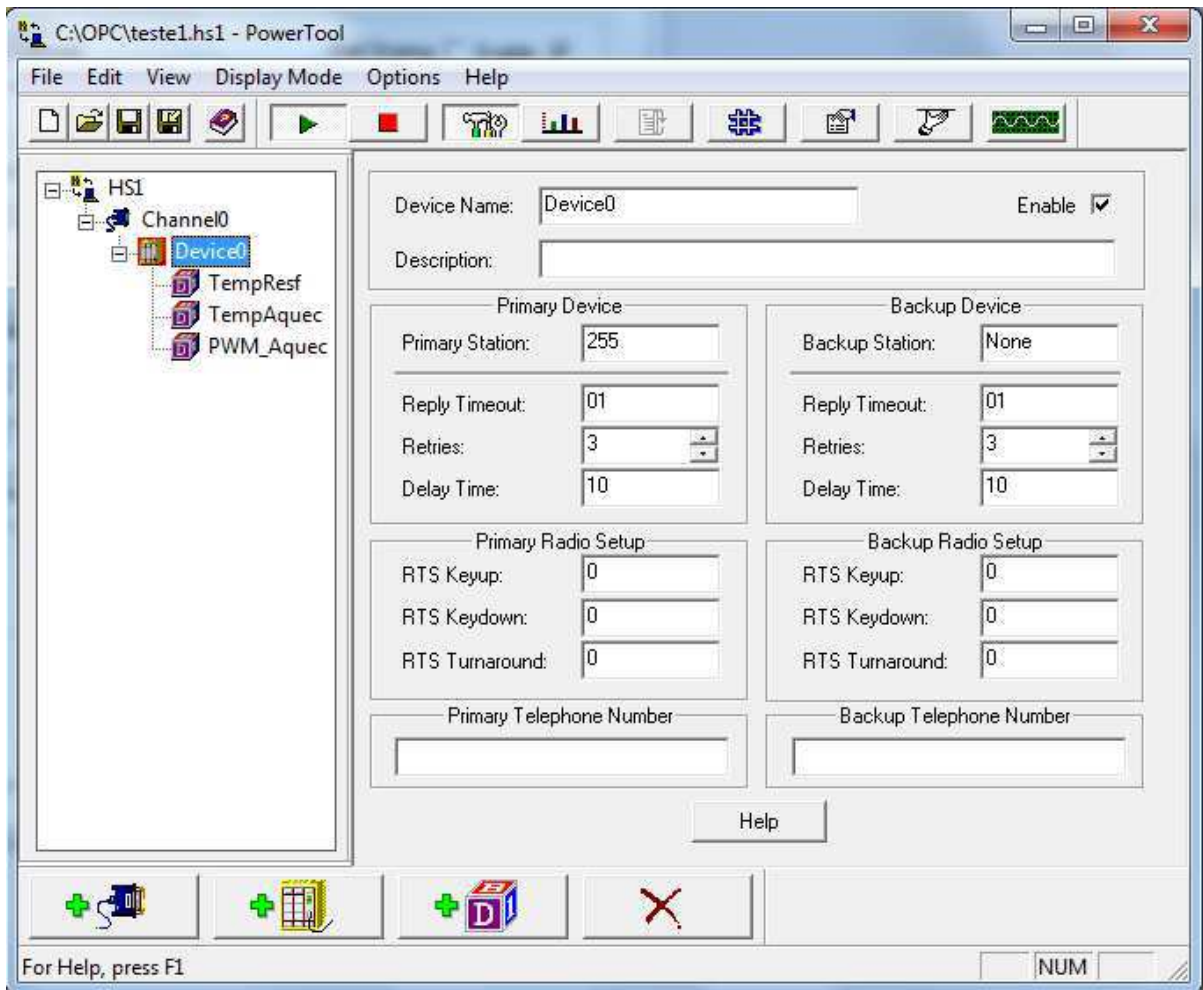


Figura 19 - Janela do programa HS1Power Tool para configuração dos parâmetros equipamento para efetuar a comunicação.

Passo 5: Criar os *DataBlocks* necessários para a aplicação e habilitá-los. Os *DataBlocks* são os endereços das variáveis de interesse, Figura 20.

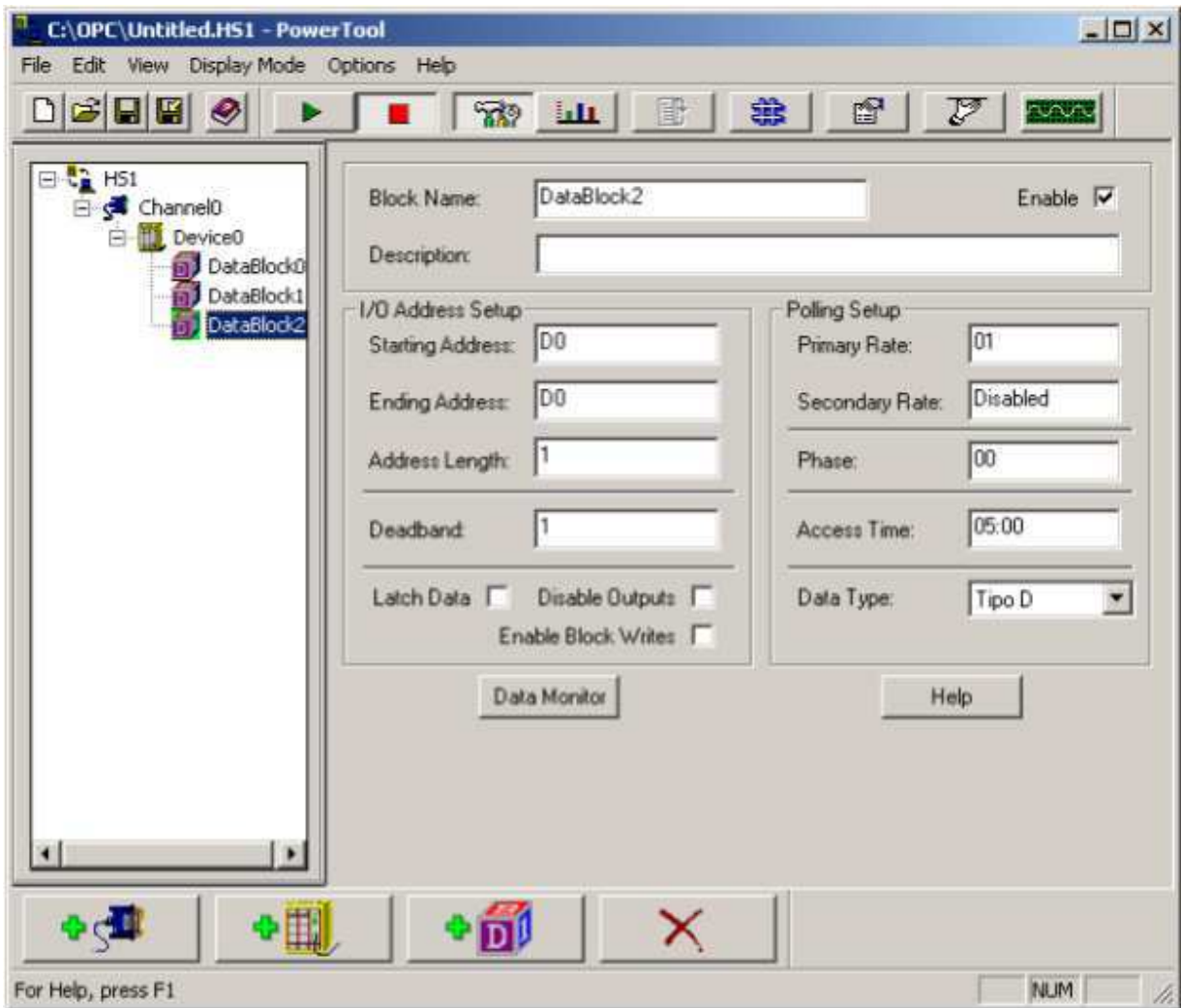


Figura 20 - Janela do programa HS1Power Tool para configuração dos parâmetros equipamento para efetuar a comunicação.
Fonte: [8]

Passo 6: Clicar no botão *PLAY* e verificar se foi realizada a comunicação clicando no botão *Statistics*.

Passo 7: Minimizar a janela do Programa HS1 Power Tool.

3.5 COMUNICAÇÃO MATLAB/CLP

Após criação da interface para comunicação do MATLAB com o CLP, basta criar um programa no Simulink que faça leitura e escrita das variáveis.

Descreve-se nos seguintes passos:

Passo 1: Abrir o programa MATLAB.

Passo 2: Abrir o programa Simulink.

Passo 3: Clicar no botão new model/salvar o arquivo com no formato .slx

Passo 4: Inserir no programa e configurar o bloco OPC Configuration da OPC Toolbox.

O bloco OPC Configuration, Figura 21, define o cliente OPC que será utilizado, configura um pseudo ambiente de tempo real e define o ambiente para erros e eventos do modelo. Esse bloco não possui portas de entrada, apenas uma porta de saída opcional para exibir a latência (tempo gasto em cada passo da simulação). Não é possível utilizar mais de um bloco OPC Configuration no mesmo modelo. [9]

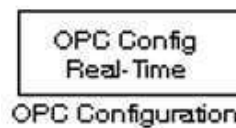


Figura 21 – Bloco OPC Configuration.

Fonte: [9]

Deve-se clicar duas vezes no bloco *OPC Config Real-Time* e selecionar o cliente OPC, na janela indicada na Figura 22, criado pelo software HS1 clicando de forma sequencial em:

Configure OPC Clients/ Add/ select/ Intellution.HS1OPC/OK/ok/close/OK

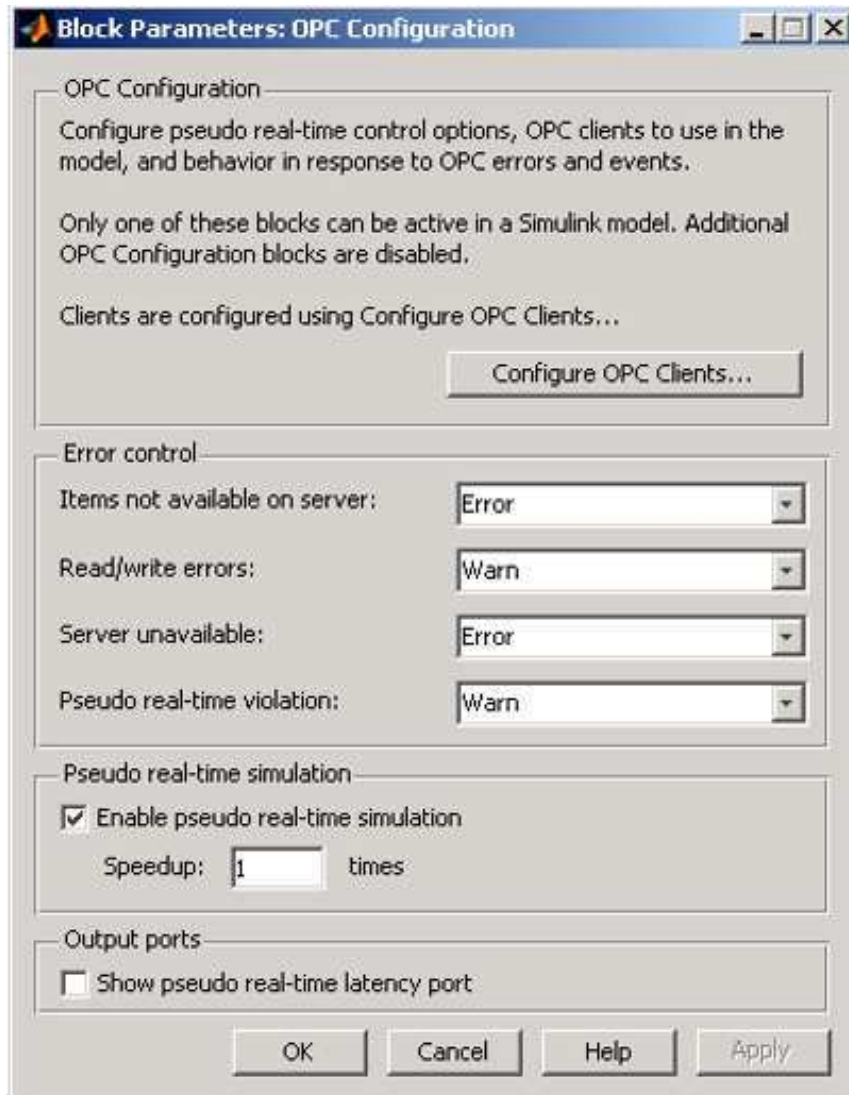


Figura 22 – Janela para configuração do bloco OPC configuration.
Fonte: [9]

Passo 5: Inserir no programa e configurar o bloco OPC read da OPC Toolbox, Figura 23.

Esse bloco possui três saídas: values (V), quality ID's e time stamps (T). A saída utilizada na aplicação foi apenas a value (V), que recebe o valor lido pela variável de interesse.

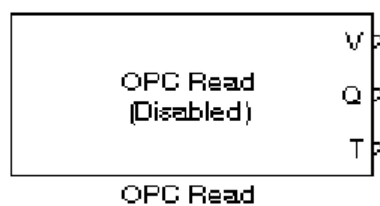


Figura 23 – Bloco OPC Read.
Fonte: [9]

O bloco OPC Read lê dados de um ou mais itens de um servidor OPC. A operação de leitura pode ser de forma síncrona (sincronizada com a cache ou com o dispositivo) ou assíncrona (do dispositivo). [9]

Após clicar duas vezes no bloco OPC Read, abrir-se-á a janela da Figura 24.

Para selecionar as variáveis para medição, basta seguir de forma sequencial clicando em:

Add Items/ Device0/ selecionar M15, D15 e D12/ >> / OK/OK

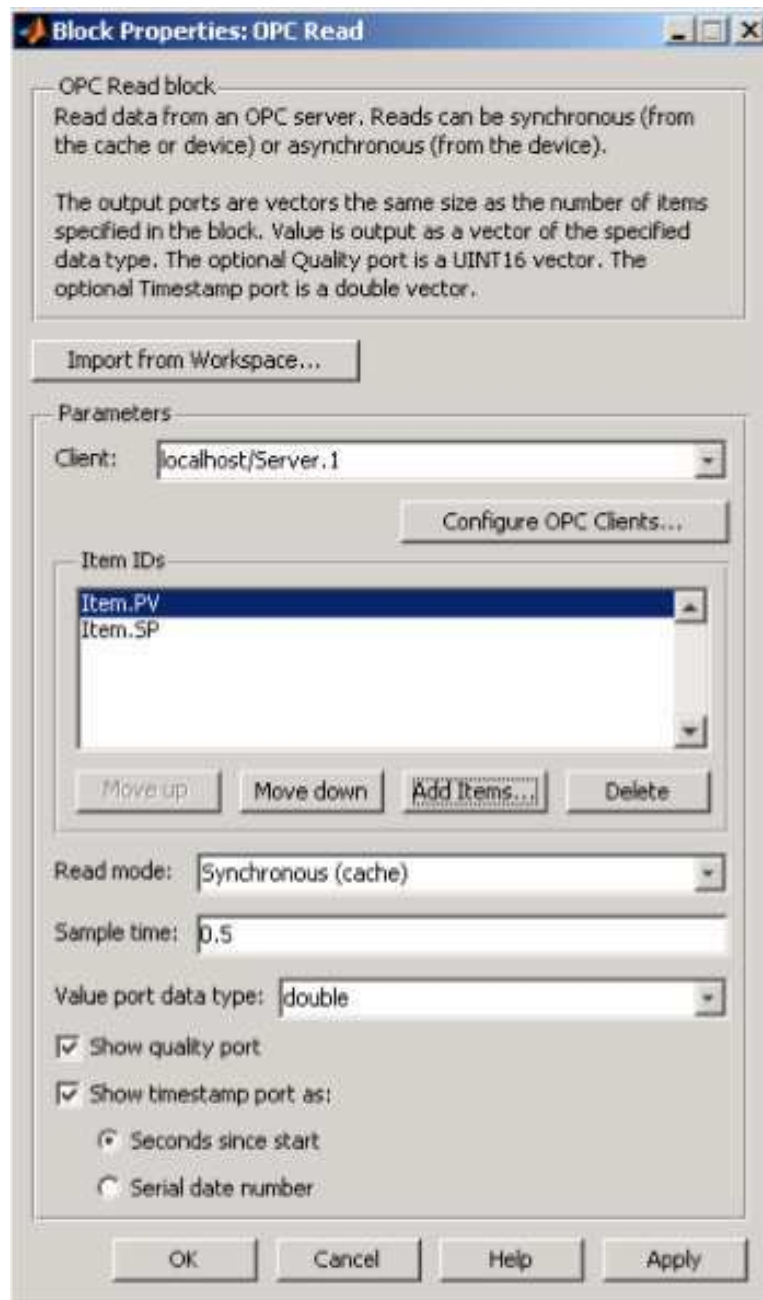


Figura 24 - Janela para configuração do bloco OPC Read.
Fonte: [9]

Passo 6: Inserir no programa e configurar o bloco OPC Write da OPC Toolbox, Figura 25.

O bloco OPC Write escreve valores em um ou mais itens do servidor OPC.

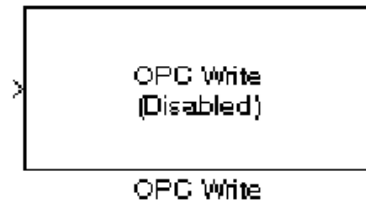


Figura 25 - Bloco OPC Write
Fonte: [9]

Após clicar duas vezes no bloco OPC Write, abrir-se-á a janela da Figura 26.

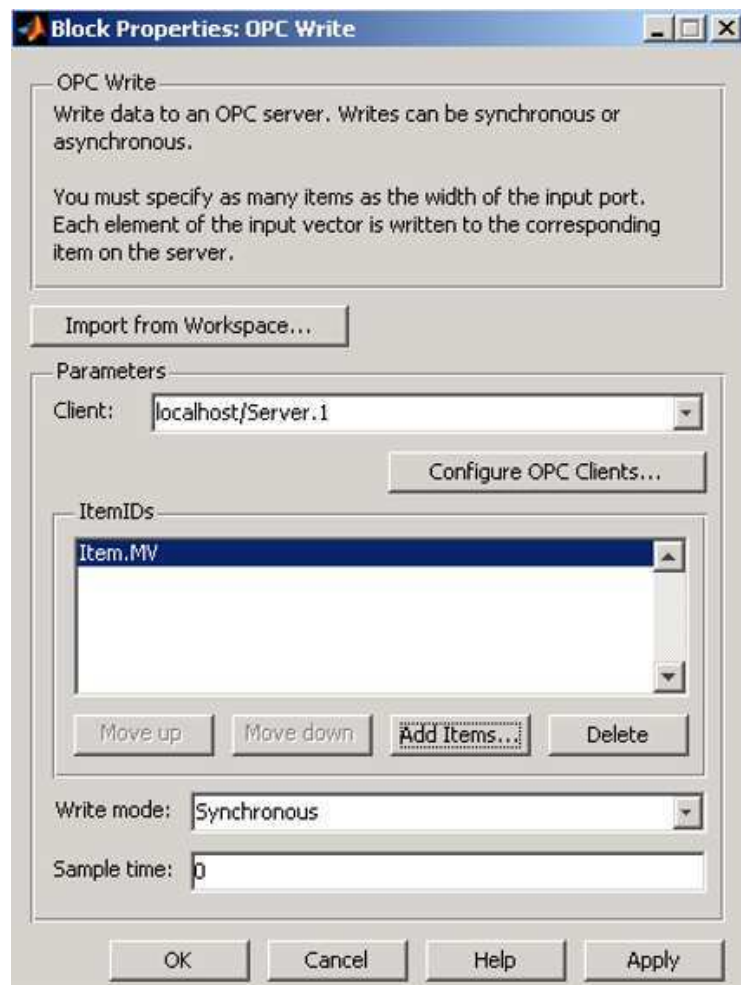


Figura 26 - Janela para configuração do bloco OPC Write.
Fonte: [9]

De forma análoga ao bloco OPC Read, para selecionar a variável para escrita, basta seguir de forma sequencial clicando em:

Add Items/ Device0/ selecionar M15/ >> / OK/OK

Passo 7: Na biblioteca do Simulink, pode-se buscar por blocos. Acredita-se que não há necessidade de detalhar os blocos desse passo, por considerá-los mais comumente utilizados. Adicionar os blocos Time Scope, Step e To Workspace e interligá-los

O modelo criado para a identificação do sistema do módulo de aquecimento assemelha-se ao da Figura 27. No modelo da aplicação não havia apenas um degrau na entrada de M15, mas, fez-se uma varredura nos valores de 0 % a 100% do dutycycle do PWM. Para isso, utilizou-se somadores e subtratores intercalados de 400 s.

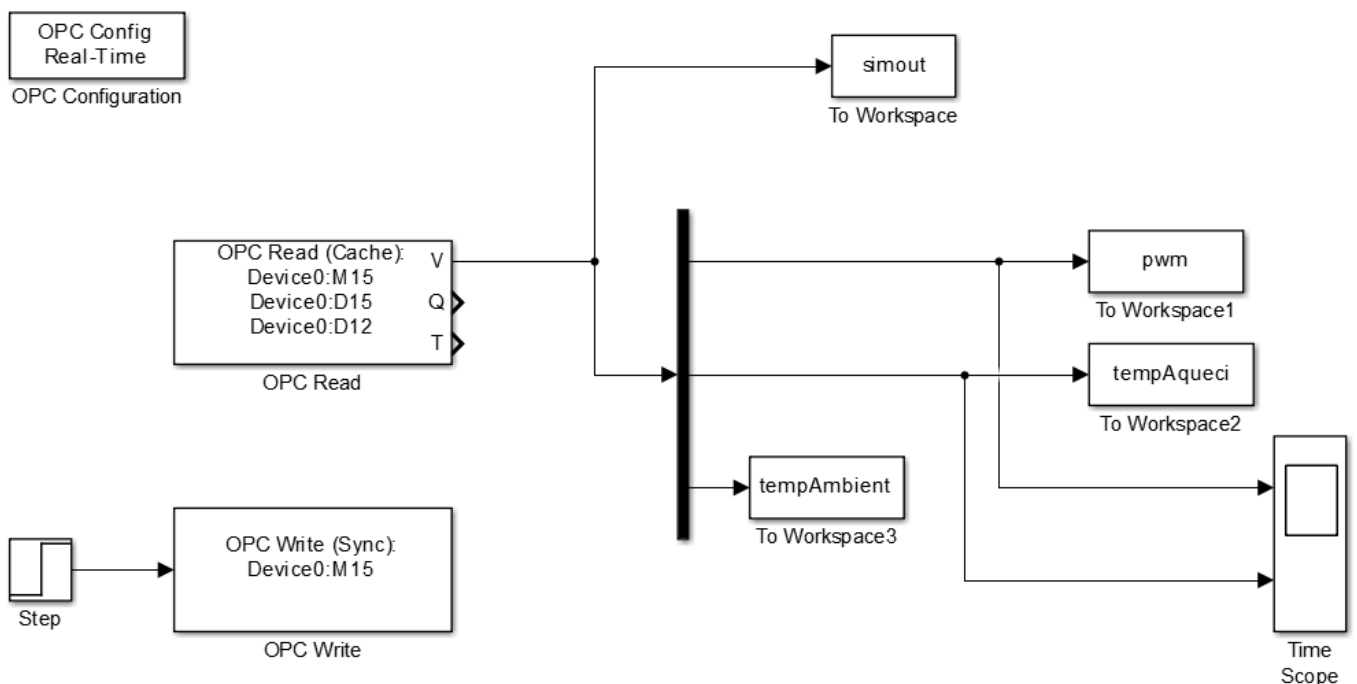


Figura 27 – Programa no Simulink para escrita e leitura de variáveis para identificação do sistema do módulo de aquecimento.

Fonte: [9]

Passo 8: Com o programa todo montado, basta iniciar o experimento. A esteira deve estar ligada e conectada ao CLP, previamente programado. Então, clica-se no botão play.

Após conclusão do experimento, obteve-se o resultado apresentado na Figura 28 e gerou-se um arquivo com os vetores das variáveis M15 e D15. No qual M15 é o valor do dutycycle do PWM, em azul, e D15 é a temperatura do módulo de aquecimento, em vermelho.

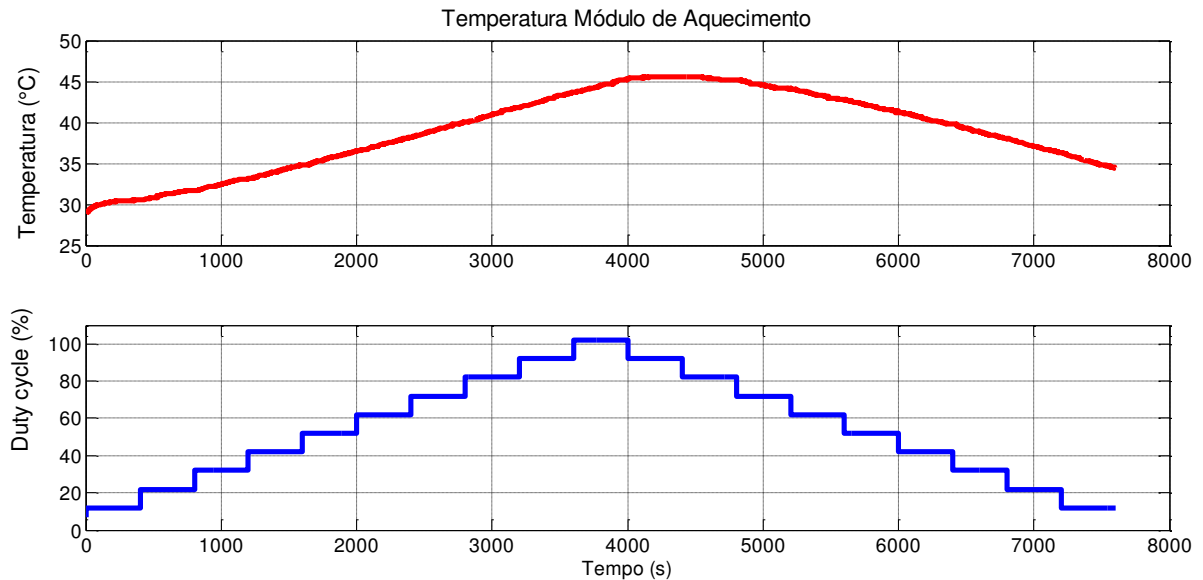


Figura 28 - Gráfico da temperatura do módulo de aquecimento (°C) para diversos degraus aplicados ao dutycycle do PWM da resistência em função do tempo.

3.6 IDENTIFICAÇÃO DO SISTEMA DE TEMPERATURA

Utilizou-se a Toolbox System Identification [10] para fazer a identificação do sistema e obter uma função de transferência simplificada para o sistema.

Como pode-se ver na Figura 29, foram determinadas diversas funções de transferência para diferentes valores de pólos e zeros buscando obter a mais simples que mais se aproximasse da real.

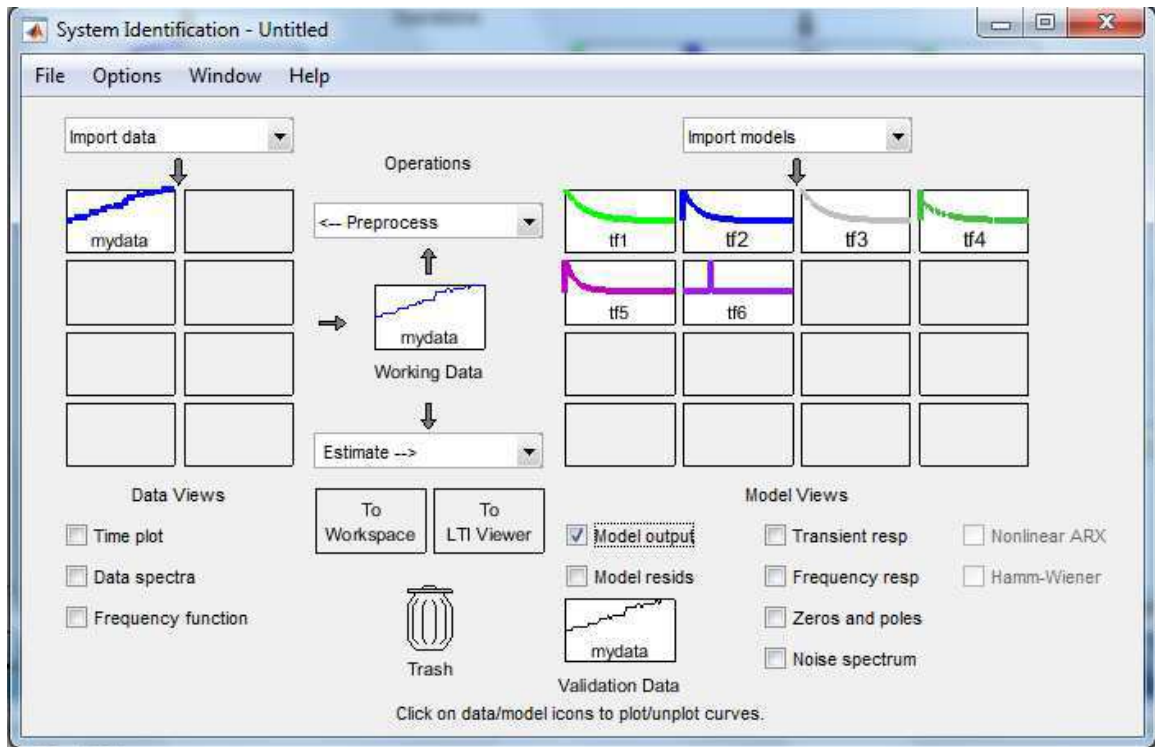


Figura 29 - Janela inicial do System Identification.

Como mostra a figura 30, para gerar as funções de transferência, após selecionar os dados obtidos no experimento citado no tópico anterior, configurou-se a janela de parâmetros da função de transferência.

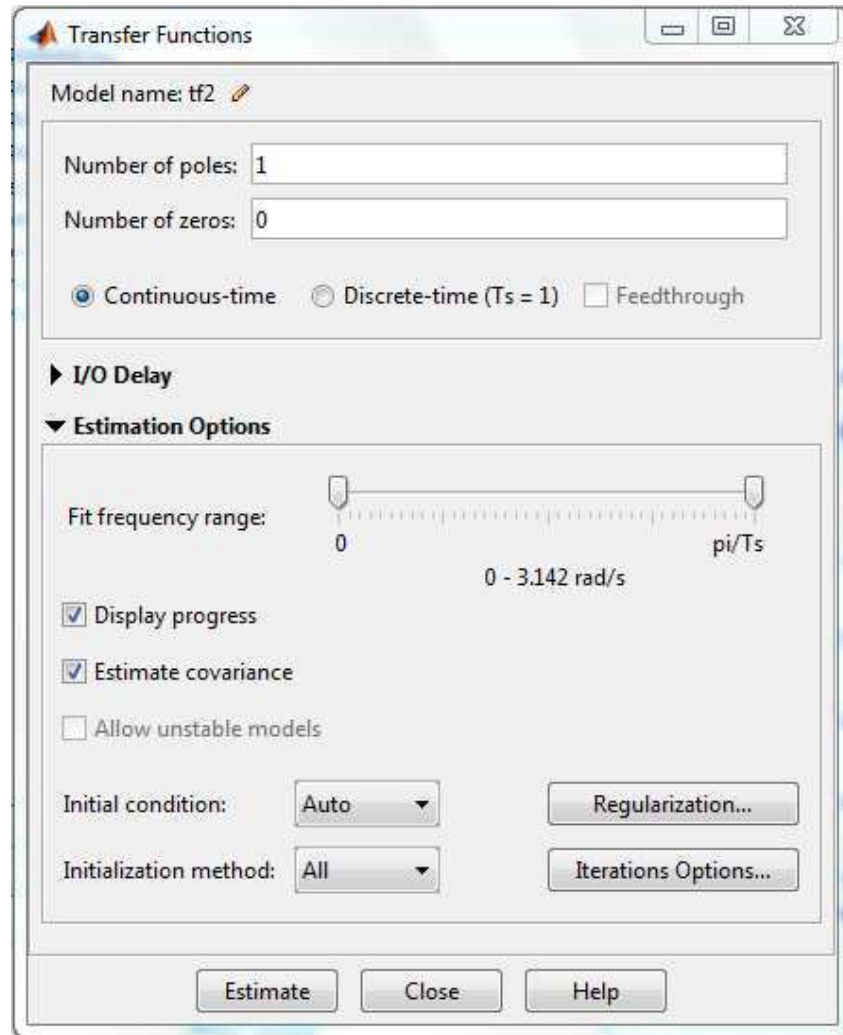


Figura 30 - Janela para ajuste dos parâmetros da Função de Transferência.

Da Tabela 5, observamos que, para as diferentes combinações realizadas, o resultado que apresentou o melhor *fitting* foi a tf5 com três pólos e um zero.

Tabela 5 - Fitting das Funções de Transferência para diversos valores de pólos e zeros.

Função de Transferência	Número de Pólos	Número de Zeros	Fit
Tf1	1	0	27,94 %
Tf2	2	0	27,95 %
Tf3	2	1	88,27 %
Tf4	3	0	25,35 %
Tf5	3	1	97,31 %
Tf6	3	2	88,85 %

Para melhor visualização das funções de transferência obtidas por simulação, plotou-se todas num mesmo gráfico, como pode ser visto na Figura 31.

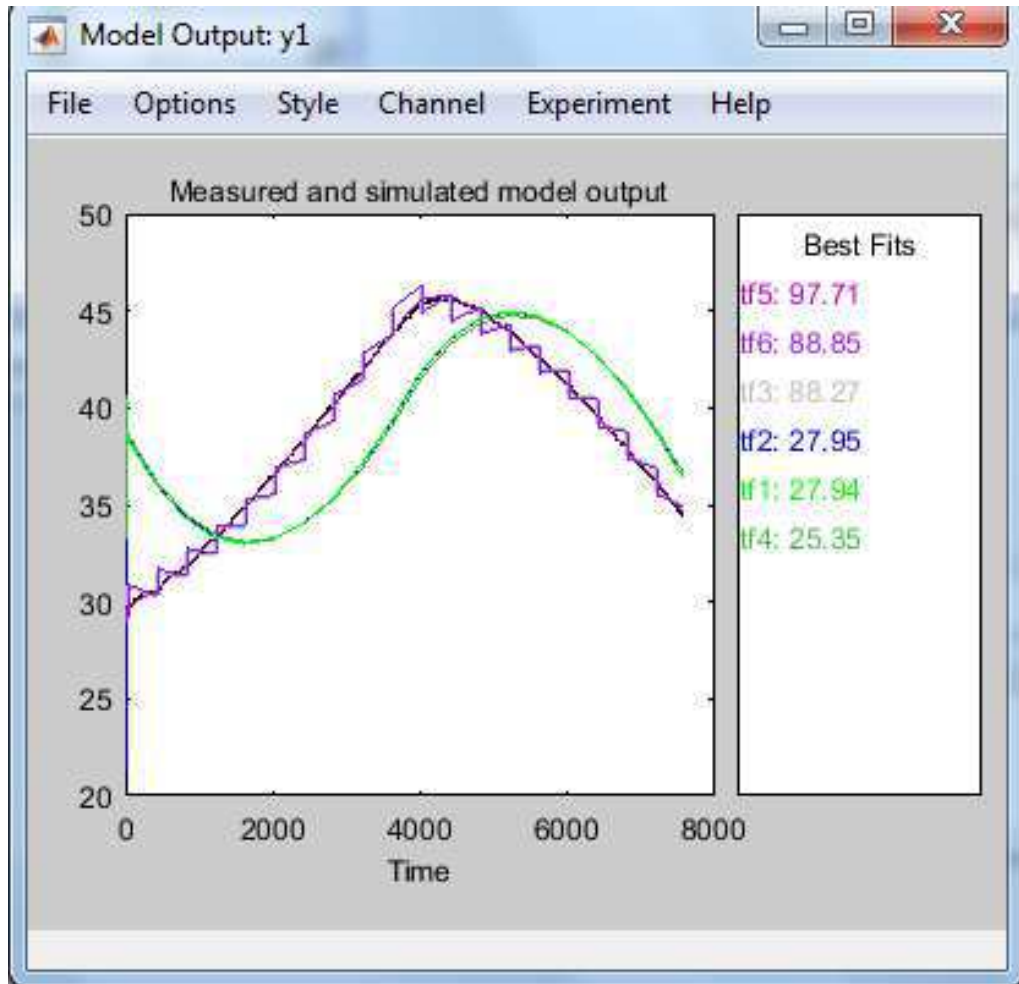


Figura 31 - Gráfico com todas as funções de transferência geradas.

A equação da função de transferência de melhor *fitting*, *tf5*, pode ser vista na Figura 32.

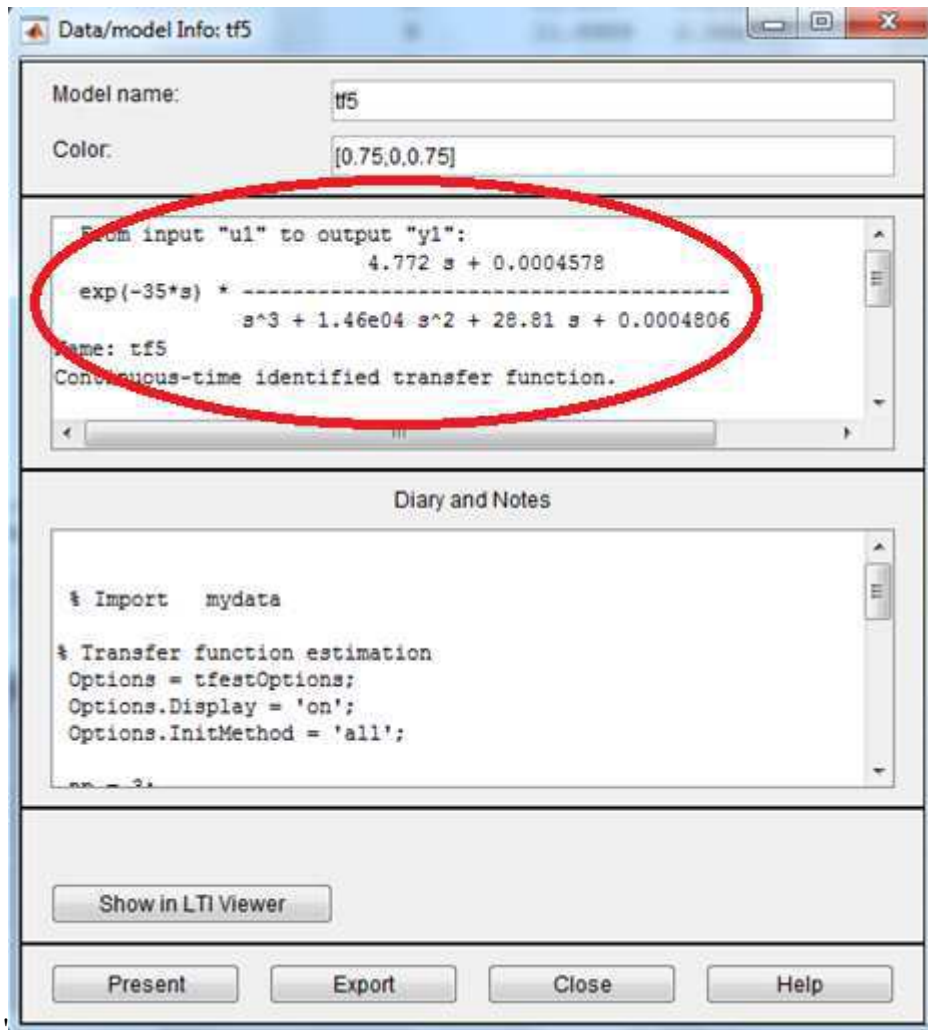


Figura 32 - Equação da FT5.

Após isso, bastou-se segurar e arrastar a *ft5* da Figura 29 para o local *To workspace* para salvar o arquivo com as informações da função de transferência.

3.7 PID TUNING

Com o arquivo com as informações da função de transferência que mais se aproximou do resultado real, pode-se obter um controlador para o sistema.

Com a toolbox do MATLAB PID Tuner [11] foi possível determinar um controlador para a temperatura do módulo de aquecimento. Apresenta-se na Figura 33 um exemplo de controlador para o modelo identificado.

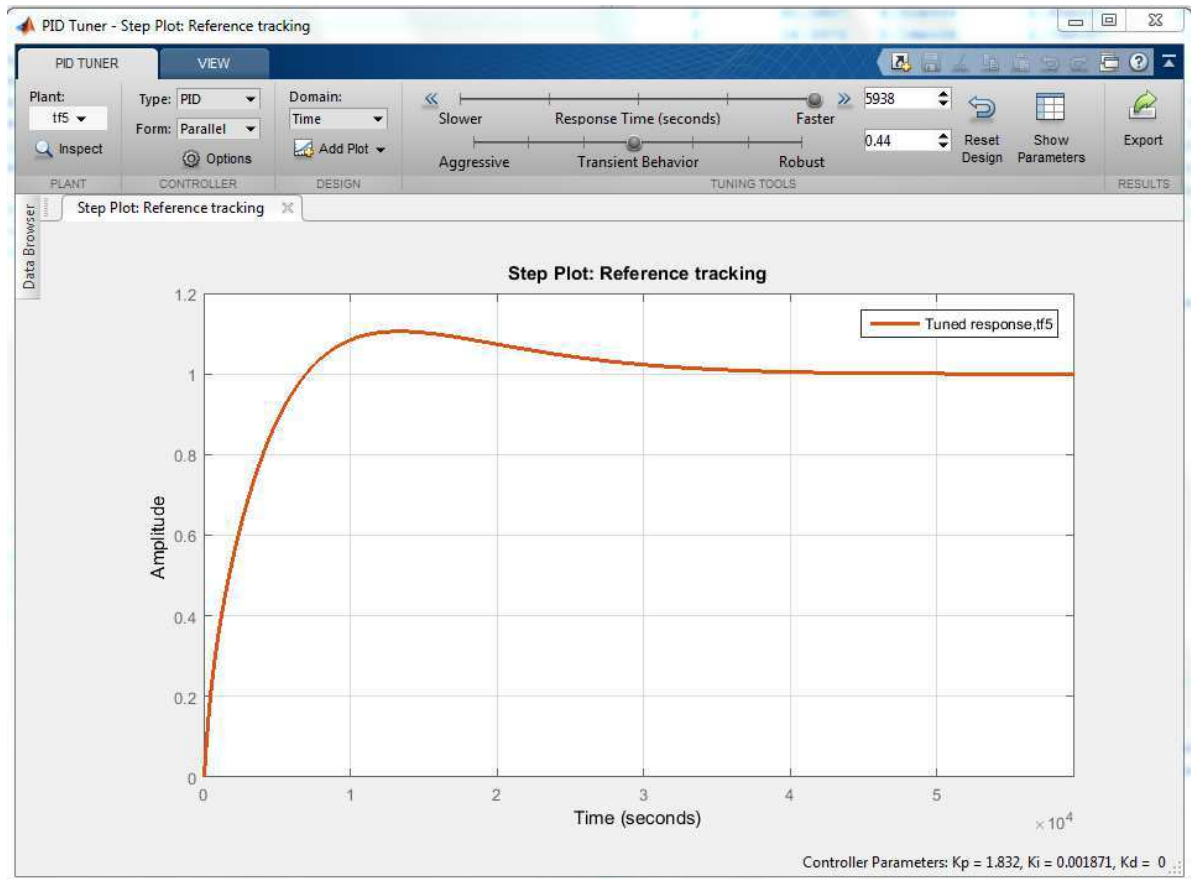


Figura 33 – Controlador PID gerado pela toolbox PID Tuner para a função de transferência para temperatura do módulo de aquecimento.

Devido à falta de tempo, não foi possível aplicar o controlador em malha fechada e observar se o resultado foi satisfatório.

Então, deixou-se de sugestão para trabalhos futuros, aplicar em malha fechada diversos PID's com diferentes valores de *Response Time* e *Transient Behavior*.

3.8 MONITORIA DE SISTEMA DE AUTOMAÇÃO INDUSTRIAL

Uma das atividades desempenhadas durante o estágio foi o acompanhamento dos alunos da disciplina Sistema de Automação Industrial, ministrada pelo professor George Acioli Júnior.

Auxiliando os alunos com problemas de lógica da linguagem Ladder e funcionamento do CLP da família ZAP 91x e ZAP 900 da HI Tecnologia.

4 CONCLUSÃO

A principal contribuição desse trabalho foi a identificação de um modelo do sistema que pode ser usado para projetar controladores para o módulo de temperatura da esteira transportadora.

Foi fundamental o estudo da linguagem Ladder, tecnologia utilizada na programação de CLP's, equipamentos amplamente utilizados em aplicações industriais. Com esse conhecimento, foi possível desenvolver diversas rotinas para teste do funcionamento dos dispositivos conectados à esteira.

Além disso, a utilização do software MATLAB, também comumente utilizado em aplicações de engenharia, foi de grande valia para leitura e escrita de variáveis, identificação do sistema e obtenção de controladores PID.

4.1 SUGESTÕES PARA TRABALHOS FUTUROS

Como opção para possíveis trabalhos futuros utilizando a esteira transportadora, sugere-se:

- Obter diversos controladores PID para o sistema identificado;
- Realizar ensaios em malha fechada para todos os controladores obtidos;
- Desenvolver um programa que realize uma sequência de procedimentos simulando uma aplicação real.

REFERÊNCIAS

- [1] Prédio do LIEC – Portal do LIEC. Disponível em: <<http://liec.ufcg.edu.br/Default.cshtml>>, Acesso em 07/06/2016.
- [2] BATISTA, E. S. Instrumentação e Controle de Esteira Transportadora Utilizando CLP. Universidade Federal de Campina Grande (Trabalho de Conclusão de Curso). Campina Grande. 2015.
- [3] BRYAN, L. A.; BRYAN, E., A. Programmable Controllers - Theory and Implementation, 2ª ed. Atlanta: Industrial Text Company, 1997.
- [4] Kit ZAP 91x- Portal da HI Tecnologia. Disponível em: <<http://www.hitecnologia.com.br/produtos/hardware/kit-de-treinamento/kit-familia-zap91x/ztk910>>
- [5] SPDSW- Portal da HI Tecnologia. Disponível em: <http://www.hitecnologia.com.br/software/controle_de_processos/spdsw>, Acesso em 07/06/2016.
- [6] Nubotics WheelWatcher - WW-12. Disponível em: <http://nubotics.com/products/ww12/ww12_manual.pdf>. Acesso em 06/06/2016.
- [7] PET10800100-ZAP900-Controlador Lógico Programável ZAP900/901. Disponível em: <<http://www.hitecnologia.com.br/repositorio/manuais-pet/familia-zap900/PET10800100.pdf>>, Acesso em 06/06/2016.
- [8] ENA0003300- HS1-OPC. Disponível em: <<http://www.hitecnologia.com.br/software/softwaresoletos/hs1-opc>>, Acesso em 07/06/2016.
- [9] MATHWORKS, Simulink, Disponível em: <<http://www.mathworks.com/products/simulink/>>, Acesso em 07/06/2016.
- [10] MATHWORKS, System Identification Tool, Disponível em: <<http://www.mathworks.com/products/sysid/>>, Acesso em 07/06/2016.
- [11] MATHWORKS, PID Tuner, Disponível em: <<http://www.mathworks.com/discovery/pidtuning.html>>, Acesso em 07/06/2016.