

RODRIGO MALVAZI CORDER

**CONTROLE EM TEMPO REAL VIA
MATLAB/SIMULINK**

Trabalho de Conclusão de Curso apresentado à
Escola de Engenharia de São Carlos, da
Universidade de São Paulo

Curso de Engenharia Elétrica com Ênfase em
Automação e Energia

Orientador: Profa. Dra. Vilma Alves de Oliveira

São Carlos
2009

Agradecimentos

À Deus, por estar presente em todos os momentos da minha vida, por guiar meus caminhos e me conceder sabedoria e saúde;

À Universidade de São Paulo (USP), por oferecer toda a estrutura necessária para o desenvolvimento deste trabalho;

À FIPAI, pelo apoio financeiro e pela bolsa concedida;

À minha orientadora Profa. Vilma Alves de Oliveira por ter me concedido a oportunidade de realizar este trabalho, pela orientação e pela amizade;

A minha família por todo esforço que realizaram para que eu chegasse até aqui;

Aos docentes e funcionários do Departamento de Engenharia Elétrica da Escola de Engenharia de São Carlos;

À todos aqueles que direta ou indiretamente contribuíram para a realização deste trabalho.

Lista de Figuras

FIGURA 2.1	<i>Real-time workshop.</i>	4
FIGURA 2.2	<i>Modo external.</i>	5
FIGURA 2.3	<i>Configuração do Simulink: Real-Time Workshop.</i>	6
FIGURA 2.4	<i>Bloco de analog input.</i>	7
FIGURA 2.5	<i>Configuração do Simulink: hardware implementation.</i>	7
FIGURA 2.6	<i>Instalação de novo hardware.</i>	8
FIGURA 2.7	<i>Abrindo o External Mode Panel.</i>	9
FIGURA 2.8	<i>Configuração de saída gráfica.</i>	10
FIGURA 2.9	<i>Iniciar Real-Time Workshop.</i>	10
FIGURA 3.1	<i>Comunicação serial.</i>	13
FIGURA 3.2	<i>Comunicação por rede.</i>	13
FIGURA 3.3	<i>Configuração do compilador C.</i>	15
FIGURA 3.4	<i>Gravando disco com o kernel.</i>	20
FIGURA 3.5	<i>xPC Target no PC-alvo.</i>	21
FIGURA 3.6	<i>Configuração de parâmetros de tempo real: Real-Time Workshop.</i>	23
FIGURA 3.7	<i>Configuração de parâmetros de tempo real: xPC Target options.</i>	23
FIGURA 3.8	<i>PC-alvo carregado com a aplicação xpc_osc3.</i>	24
FIGURA 3.9	<i>Interface xPC Target no PC-alvo.</i>	25
FIGURA 4.2	<i>Senóide de 40 Hz.</i>	28

FIGURA 4.3	Senóide de 200 Hz.	28
FIGURA 4.4	Onda quadrada de 40 Hz.	28
FIGURA 4.5	Onda quadrada de 200 Hz.	28
FIGURA 4.1	<i>Entrada e saída analógica.</i>	28
FIGURA 4.6	Senoide de 40 Hz.	29
FIGURA 4.7	Onda quadrada de 40 Hz.	29
FIGURA 4.8	<i>Controle de um motor CC através do simulink.</i>	29
FIGURA 4.9	<i>Blocos de entrada e saída analógica.</i>	30
FIGURA 4.10	<i>Diagrama esquemático do sistema de controle do motor CC.</i>	31
FIGURA 4.11	<i>Tensão do taco gerador do motor CC.</i>	31
FIGURA 4.12	<i>Simulação de um motor CC através do xPC Target.</i>	32
FIGURA 4.13	<i>Velocidade de um motor CC visualizada no PC-alvo.</i>	33
FIGURA 4.14	<i>Simulação de um motor CC com ruído.</i>	33

Lista de Tabelas

TABELA 3.1	Placas de rede compatíveis com o produto xPC Target	17
TABELA 3.2	Placas de rede compatíveis com o produto xPC Target	19

Resumo

Este trabalho tem por objetivo estudar as possibilidades de implementar controle em tempo real via Matlab/Simulink. Foi realizado o estudo de duas ferramentas pertencentes ao aplicativo Matlab: Real-Time Workshop e xPC Target. A primeira permite o controle de plantas reais e a segunda, a partir de um PC-host, permite a monitoração e o controle de plantas reais conectadas em um PC-alvo. Foram realizadas algumas aplicações para confirmar a utilidade das ferramentas. Primeiramente, um programa foi elaborado apenas para enviar e receber sinais que puderam ser visualizados de forma gráfica. Em adição, foi implementado o controle da velocidade de um motor de corrente contínua via Real-Time Workshop e por fim, foi realizada uma aplicação com o xPC Target para a visualização da velocidade de um motor de corrente contínua no PC-alvo.

Palavras chave: Matlab, Simulink, controle, real-time workshop, xPC target, plantas reais.

Abstract

This work has the objective of studying the possibilities of real-time control implementing by Matlab / Simulink. It was studied two tools within the Matlab software: Real-Time Workshop and xPC Target. The first allows the control of real plants and the second from a host PC, allows the monitoring and control of real plants connected to a target PC. Some applications were performed to confirm the usefulness of the tools. First, a program was developed only to send and receive signals that could be viewed in graphical form. In addition, the speed control of a direct current motor was implemented and finally, the xPC Target was used to display the direct current motor speed response to a step input.

Keywords: Matlab, Simulink, control, real-time workshop, xPC target, real plants.

Sumário

Lista de Figuras	vii
Lista de Tabelas	ix
Resumo	xi
Abstract	xiii
1 Introdução	1
2 Real-time Workshop	3
2.1 Visão Geral	3
2.2 Processo do Real-time Workshop	3
2.3 Configuração do Matlab/Simulink	4
2.4 Instalação de Hardware Externo	6
2.5 Configuração de Saídas Gráficas	8
2.6 Criando o Modelo em Tempo Real	9
3 xPC Target	11
3.1 Visão Geral	11
3.2 Características do xPC Target	12
3.2.1 Real-Time do Kernel	12
3.2.2 Target Boot Disk	12
3.3 Conexão entre PC-host e PC-alvo	12
3.4 Configuração do Compilador C para o xPC Target	13
3.5 Comunicação entre PC-host e PC-alvo	14
3.6 Processo do xPC Target	15
3.7 Comunicação por Rede	16
3.7.1 Vantagens da Comunicação em Rede	16
3.7.2 Placas de Rede	16

3.8	Propriedades para Comunicação TCP/IP	18
3.9	Opções do xPC Target Boot	19
3.10	Iniciando o xPC Target	21
3.10.1	Iniciando o PC-alvo	21
3.10.2	Configurando Parâmetros de Tempo-real	21
3.10.3	Carregando a Aplicação Alvo	24
3.10.4	Iniciando o xPC Target	24
4	Aplicações Realizadas	27
4.1	Aplicação Real-Time Workshop	27
4.2	Aplicação xPC Target	32
5	Conclusão	35
	Referências Bibliográficas	37

Capítulo 1

Introdução

Implementação de controladores em tempo real é uma tarefa que leva muito tempo em qualquer processo de controle devido ao fato da complexidade e do grande número de tarefas que devem ser realizadas (Teng 2000). Esses controladores podem ser aplicados em diversas áreas como robótica, automação de processos industriais, na indústria aéro espacial e em sistemas de transportes (Eker e A.Cervin 1999). Tendo em vista isto, novas ferramentas estão sendo estudadas e uma destas é o Real-Time Workshop (RTW), que é apresentado neste projeto. Com esta ferramenta, sinais de plantas reais são enviados para o Matlab através de uma placa de aquisição de dados e com isso calcula-se o sinal de controle a ser aplicado de volta ao processo.

O RTW é uma extensão do Simulink e Matlab, a qual gera e compila códigos em C, proveniente de modelos do Simulink. Para sua utilização, Matlab e Simulink devem ser configurados corretamente. Esta configuração foi realizada e é apresentada neste trabalho.

Além do RTW, este trabalho apresenta a ferramenta xPC Target. Esta ferramenta permite o controle e supervisão de plantas reais conectadas a um PC, através de outro PC que não necessita estar próximo ao primeiro, ou seja, o controle é realizado a distância.

Para utilizar o RTW e o xPC Target e verificar a sua eficiência, foi escolhido como aplicação o controle de um motor de corrente contínua. Pode-se constatar que as ferramentas foram eficientes e atingiram as expectativas esperadas.

Este trabalho está dividido em 6 capítulos. O Capítulo 2, seguinte a esta introdução,

aborda as configurações necessárias para realizar o controle em tempo real utilizando Matlab e Simulink. O Capítulo 3 apresenta o produto xPC Target e a configuração necessária para a sua utilização. O Capítulo 4 ilustra as aplicações realizadas em tempo real através do xPC Target. Finalmente, o Capítulo 5 apresenta a conclusão do projeto e futuros estudos a serem realizados.

Capítulo 2

Real-time Workshop

2.1 Visão Geral

O projeto contempla inicialmente o estudo de ambientes de tempo real em Matlab para prototipagem e simulação de sistemas de controle com o processo físico na malha. O Real-Time Workshop, uma extensão do Simulink e Matlab, gera e compila códigos executáveis, proveniente de modelos do Simulink, podendo assim criar uma ampla variedade de aplicações em tempo real. Esta ferramenta fornece:

- Geração automática de códigos para uma grande variedade de plataformas
- Implementação rápida e direta
- Integração direta com Matlab e Simulink com recursos de análise e documentações
- Interface gráfica simples

2.2 Processo do Real-time Workshop

O processo realizado pelo Real-Time Workshop é ilustrado na Figura 2.1. A partir de um modelo criado no Matlab ou Simulink, a ferramenta cria um código em C ou C++ após a execução do comando *Build Model*. Este código é então compilado para gerar um código executável tornando possível a aplicação. O arquivo executável é um arquivo binário de aplicação em tempo real (*Real-Time Workshop Getting Started* 2008).

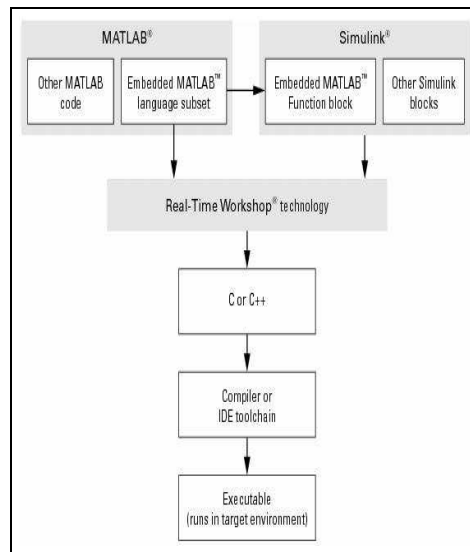


Figura 2.1: *Real-time workshop*.

Para a aplicação do RTW neste projeto foi utilizado como interface uma placa de aquisição de dados. Utilizou-se a placa *NI AT-MIO-16E-10* a qual é reconhecida pelo Simulink, permitindo assim a implementação em tempo real.

2.3 Configuração do Matlab/Simulink

Para fazer a comunicação entre Simulink e uma placa de aquisição de dados é necessário que o Simulink esteja configurado no modo *External* como ilustrado na Figura 2.2. Este modo utiliza o módulo de interface MEX-file para atualizar novos parâmetros e retornar sinais para o modelo em tempo real, podendo ser visualizados nos blocos de Scope. Os sistemas em tempo real controlam ambientes onde as restrições de tempo são bem definidas e por esta razão devem apresentar alta confiabilidade, com resultados corretos, precisos e no tempo desejado (Gómez 2004). Para isto é necessário a instalação do *Real-Time kernel*, que dará a maior prioridade de execução para o arquivo executável em tempo real, tornando-o livre de interferências durante a amostragem. Durante a execução em tempo real, o *kernel* irá interferir quando necessário para garantir que o modelo tenha prioridade no uso do CPU quando ocorre uma atualização. Quando a atualização estiver completa, o *kernel* libera o CPU para outras tarefas.

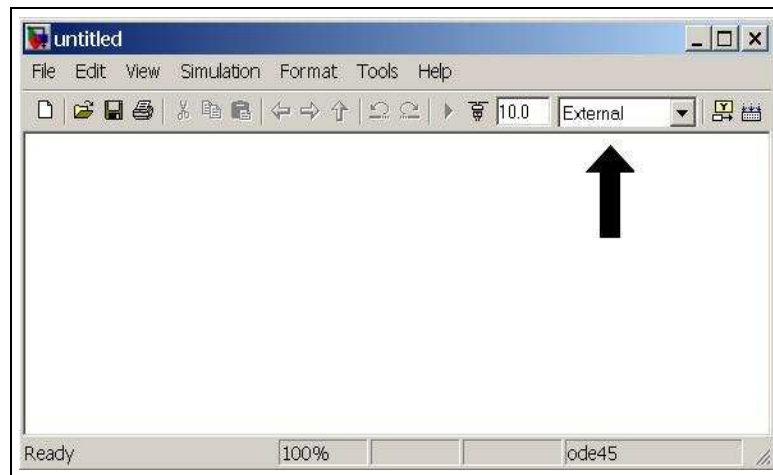


Figura 2.2: *Modo external.*

Para a instalação do *kernel*, os seguintes passos devem ser seguidos:

1. Na janela de comandos do Matlab digite: `rtwintgt -install`

2. Uma mensagem aparecerá:

You are going to install the Real-Time Windows Target kernel.

Do you want to proceed? [y] :

3. Digite *y* para instalar.

4. Uma mensagem de confirmação aparecerá:

The Real-Time Windows Target kernel has been successfully installed.

5. Reinicie seu computador.

6. Para conferir se o *kernel* foi instalado corretamente, digite: `rtwho`.

Se o *kernel* estiver instalado corretamente, informações sobre o RTW e sobre o *kernel* serão informadas como abaixo:

Real-Time Windows Target version 3.2.0 (C) The MathWorks, Inc. 1994-2008

Running on Multiprocessor APIC computer.

MATLAB performance = 100.0

Kernel timeslice period = 1 ms

Feito isto, o próximo passo é configurar o Simulink. Para dar início a esse processo, na barra de menu do Simulink abrir a janela de configurações de parâmetros em **Tools** e **Real-Time Workshop**. Escolher **rtwin.tlc**, como pode ser visto na Figura 2.3, na opção *System target file* em **Real-Time Workshop**. Com isto, o sistema poderá criar os arquivos necessários para poder rodar em tempo-real.

Além disto, nesta mesma janela de configurações, em **Hardware Implementation**, selecione **32-bit Real-Time Windows Target** em **Device type** e marque **None** em **Emulation hardware**. Clique em **Apply** para aplicar as alterações ao seu modelo e deixar a caixa de diálogo aberta e, em seguida, clique em **OK** para aplicar as alterações ao seu modelo e fechar a caixa de diálogo. A Figura 2.5 ilustra esta passagem.

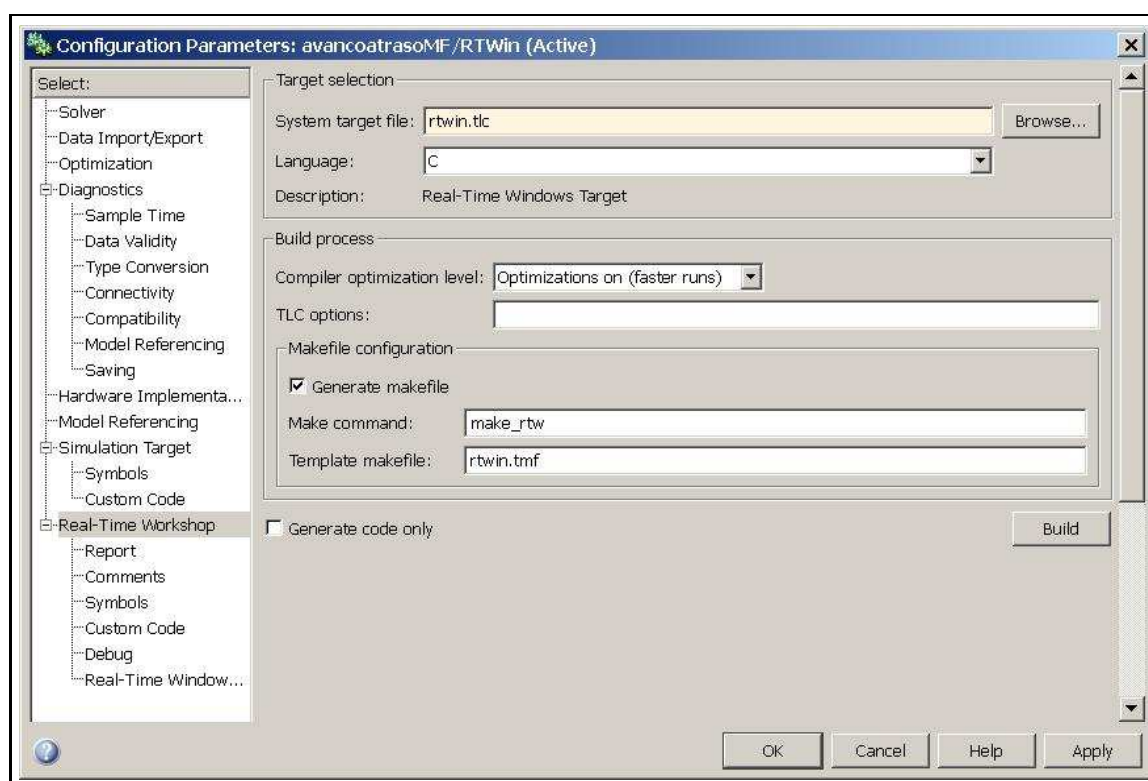


Figura 2.3: *Configuração do Simulink: Real-Time Workshop.*

2.4 Instalação de Hardware Externo

Para instalar um hardware externo adicione um bloco de **Analog Output** ou **Analog Input** (Figura 2.4), encontrados em **Real-Time Windows Target** nas opções do Simulink. Selecionando **Open Block**, uma janela de propriedades será aberta como

mostrado na Figura 2.6. Para instalar um novo hardware, a opção deve ser selecionada e um modelo, dentre os listados deve ser escolhido. Além disso, nesta mesma figura, é necessário informar o tempo de amostragem em *Sample time* e o canal que será utilizado em *Input channels*.

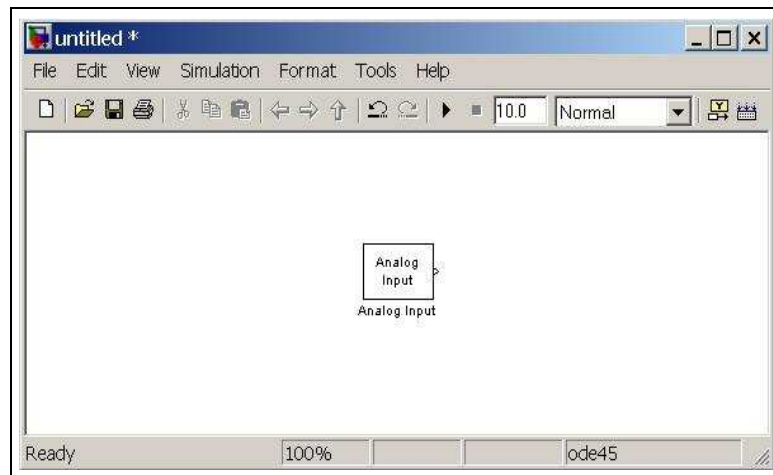


Figura 2.4: Bloco de analog input.

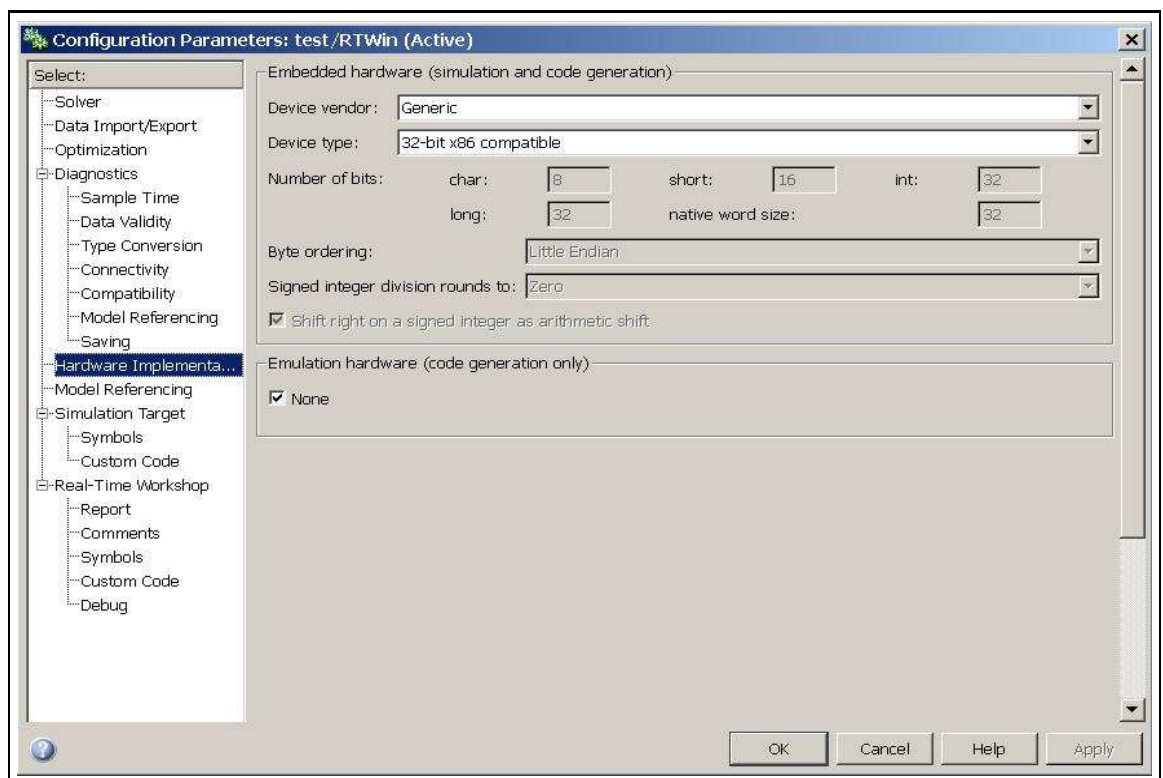


Figura 2.5: Configuração do Simulink: hardware implementation.

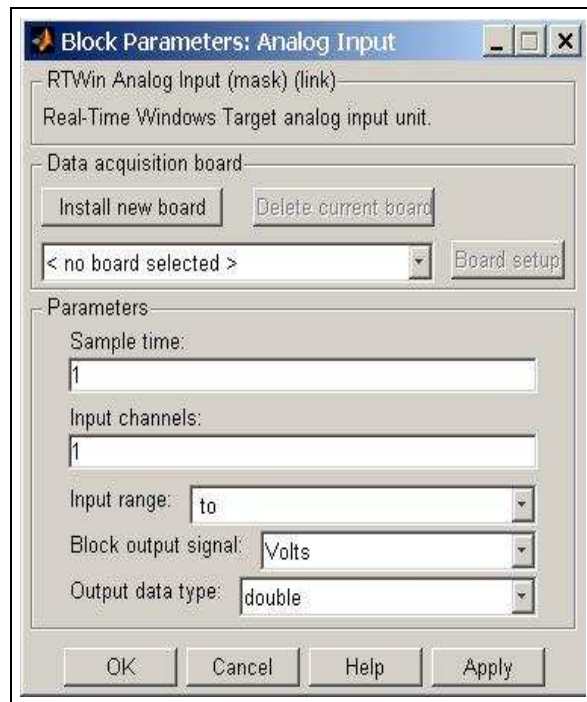


Figura 2.6: *Instalação de novo hardware.*

2.5 Configuração de Saídas Gráficas

Para a visualização dos dados recebidos pela planta em tempo real, um **Scope** deve ser configurado. Em **Tolls** selecione **External Mode Controle Panel** (Figura 2.7). Na nova janela selecione **Signal e Triggering**. Na janela **External Signal e Triggering**, como mostrado na Figura 2.8, em **Source** escolha **manual**. Em **Mode** escolha **normal**. Marque a opção **Arm when connecting to target** para iniciar a visualização quando a planta for conectada ao Simulink. Selecione **Apply** e por fim, **Close**.

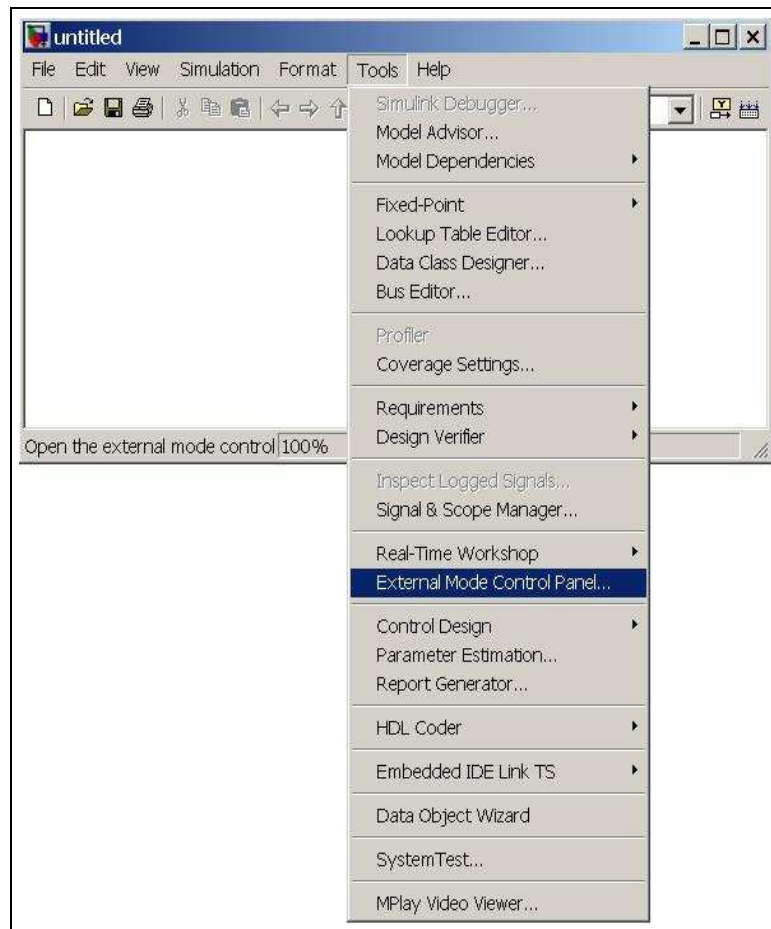


Figura 2.7: *Abrindo o External Mode Panel.*

2.6 Criando o Modelo em Tempo Real

Ao fim dos passos de configurações descritos, o modelo de tempo real pode ser criado. Para isto, após criada a planta desejada, o comando **Build Model**, em **Tools** e **Real-Time Workshop**, deve ser selecionado. Com isto, o arquivo executável é gerado. Abra **External Mode Control Panel** em **Tools** e selecione **Connect** (Figura 2.9). Por fim, selecione **Start Real-Time Code** para iniciar a aquisição de dados em tempo real.

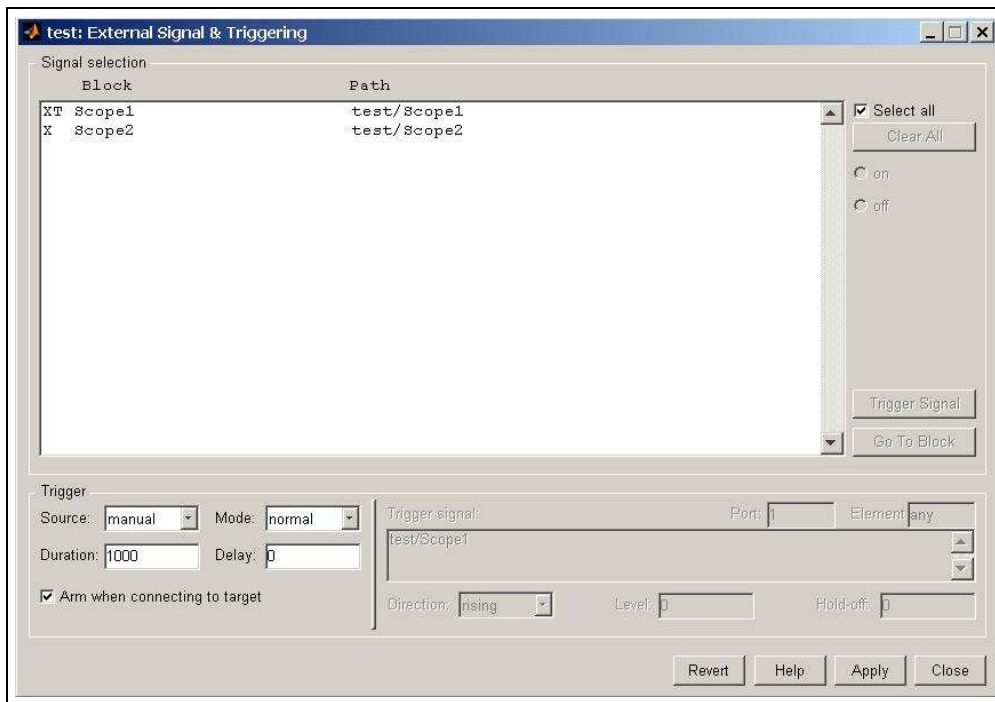


Figura 2.8: Configuração de saída gráfica.

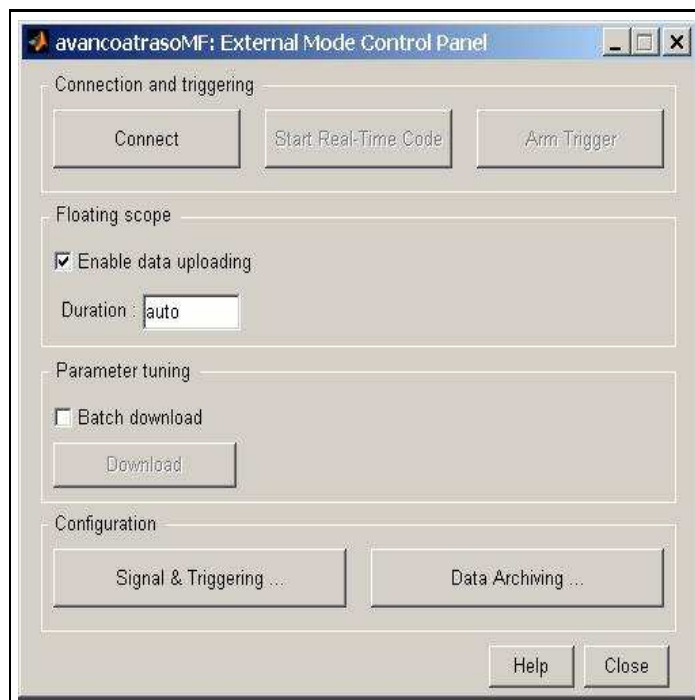


Figura 2.9: Iniciar Real-Time Workshop.

Capítulo 3

xPC Target

3.1 Visão Geral

O produto xPC Target é uma solução para protótipos, testes e implantação de sistemas de tempo real (*xPC Target Getting Started 2008*). É um ambiente que usa um computador alvo, separado de um PC host, para a execução de aplicativos em tempo real. O ambiente xPC Target inclui muitos recursos para auxiliar na construção de sistemas em tempo real. Neste ambiente é utilizado um computador desktop como um PC host com Matlab e Simulink para criar modelos. Depois de criar o seu modelo, pode-se executar simulações em tempo real a distância.

O xPC Target permite adicionar blocos de entrada e saída para o seu modelo e, em seguida, usar o PC host com o Real-Time Workshop e um compilador C / C ++ para criar código executável. O código executável é carregado a partir do PC host para o PC alvo. Depois de carregar o código executável, pode-se executar e testar uma aplicação em tempo real. Para a utilização do xPC Target há a necessidade de hardware e software específicos, os quais são listados abaixo:

- Hardware requerido - O software xPC Target requer um PC host e um PC-alvo conectados em rede e uma placa de aquisição de dados que seja reconhecida pelo xPC Target.
- Software requerido - O módulo xPC Target requer um software Microsoft Visual C / C ++ Compiler (Versão 6.0, 7.1 ou 8.0) ou Open Watcom C / C ++

Compiler (Versão 1.3). Além disso, o módulo requer Matlab, Simulink, e Real-Time Workshop software.

3.2 Características do xPC Target

3.2.1 Real-Time do Kernel

O módulo xPC Target não requer Microsoft DOS, Microsoft Windows, Linux, ou qualquer outro sistema operativo no computador alvo. Ao invés disso, inicia-se o PC alvo com um disco ou CD de inicialização que inclui o xPC Target kernel.

Depois que o *kernel* começa a rodar, uma mensagem de boas-vindas é mostrada com informações sobre a conexão host-alvo. O kernel faz com que o PC-alvo aguarde o pedido para baixar uma aplicação do PC host. O PC-alvo recebe, cópia para os endereços de suas seções designadas, e configura a aplicação de destino pronta para começar. Pode-se então utilizar outras funções e utilidades do xPC Target para se comunicar com a aplicação alvo.

3.2.2 Target Boot Disk

O boot disk elimina a necessidade de instalar aplicativo Matlab e Simulink, alterar configurações do aplicativo existente, ou acessar o disco rígido no computador de destino. Esta facilidade permite usar o PC alvo para testar aplicações em tempo real e, em seguida, quando terminar com os testes, pode-se usar o PC alvo como um computador desktop novamente. O aplicativo não está permanentemente instalado no computador alvo, a menos que isso seja uma opção.

3.3 Conexão entre PC-host e PC-alvo

O xPC Target suporta duas conexões e protocolos de comunicação entre o PC host e o PC alvo: serial e TCP/IP como descrito a seguir.

Serial - O PC host e o PC alvo são conectados diretamente com um cabo serial RS-232 usando suas portas como mostrado na Figura 3.1. Esse cabo pode ter até 5 metros de comprimento e sua taxa de transferência esta entre 1200 e 115200 baud.

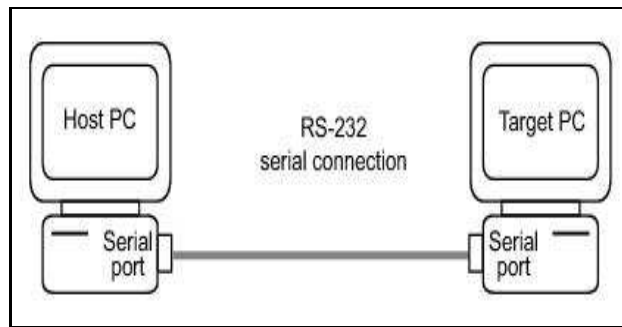


Figura 3.1: *Comunicação serial.*

TCP/IP - O PC host e o PC alvo são conectados através de uma rede como ilustrado na Figura 3.2. A rede pode ser uma LAN, ou Internet, ou uma conexão direta usando um cabo Ethernet crossover. Tanto O PC host e o PC alvo estão conectados à rede, com placas de adaptador Ethernet usando o protocolo TCP / IP para a comunicação. A taxa de transferência de dados pode ser de 10 megabits / segundo, 100 megabits por segundo, ou 1 gigabit por segundo.

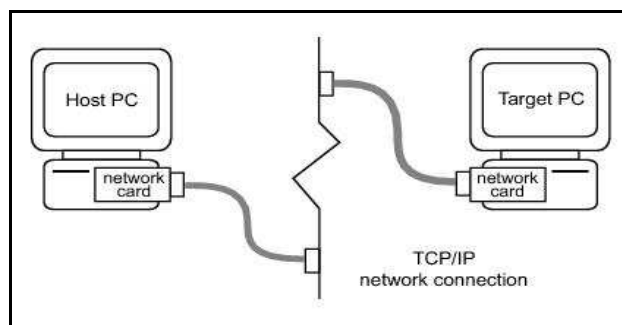


Figura 3.2: *Comunicação por rede.*

3.4 Configuração do Compilador C para o xPC Target

Para configurar o compilador do PC host, utilize a janela xPC Target Explorer. Seguem abaixo, os comandos para esta configuração

1. Em MATLAB Command Window, digite

```
xpcexplr
```

A janela xPC Target Explorer aparecerá.

2. Na janela xPC Target Explorer, selecione **Compiler(s) Configuration** como mostrado na Figura 3.3.

Na parte direita aparecerão os parâmetros do compilador.

3. Selecione seu compilador e o endereço do mesmo.
4. **Apply** para salvar as novas configurações.

3.5 Comunicação entre PC-host e PC-alvo

Usando uma conexão serial (RS-232) ou uma conexão de rede (TCP / IP), a informação é trocada entre o PC host e o PC alvo. Esta informação inclui:

- **Aplicações alvo** - O computador alvo carrega uma aplicação a partir do PC-host.
- **Controle** - Altera propriedades e controla a aplicação de destino. Isto inclui iniciar e parar, mudança de tempos de amostragens e de parada, e obter as informações sobre o desempenho da aplicação alvo e CPU.
- **Sinal de dados** - Envia sinal de dados a partir do computador alvo de análise a aplicação de destino após ter sido executado, ou ver sinal de dados durante a execução.
- **Valores dos parâmetros** - Transfere valores dos parâmetros para o computador alvo entre ou durante execuções.

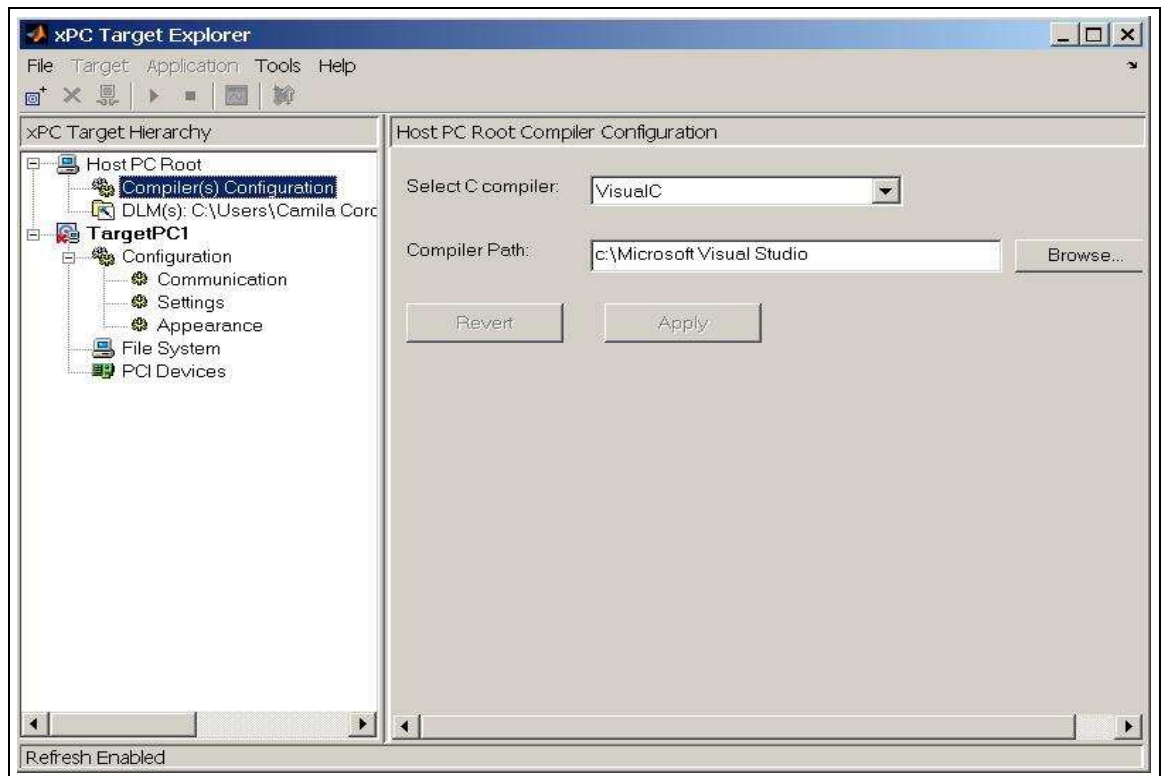


Figura 3.3: *Configuração do compilador C.*

3.6 Processo do xPC Target

A instalação do xPC Target só é necessária no PC host, ou seja, não é necessário que seja instalado no PC alvo. A seguir descreve-se o processo para projetar e construir uma aplicação em um PC host e, em seguida, executar e testar a aplicação de destino, em um PC alvo. As funções do xPC Target incluem o controle da aplicação alvo, aquisição de sinal de dados e correção dos parâmetros durante a execução em tempo real. Para este processo, as seguintes tarefas devem ser seguidas:

1. Criar um modelo no Simulink.
2. Simular o modelo.
3. Criar um aplicativo executável para o PC alvo - O Real-Time Workshop, o xPC Target e um compilador C, criam a aplicação que é executada no computador de destino. Esta aplicação em tempo real, utiliza os parâmetros iniciais do modelo Simulink que estavam disponíveis no momento da geração do código.

4. Executar a aplicação do PC alvo em tempo real - A aplicação é inicializada usando um disco carregado no PC host que carrega uma imagem do xPC Target real-time kernel. Após iniciar o PC-alvo com o disco criado no PC host, pode-se criar e carregar uma aplicação em tempo real.

O xPC Target utiliza os recursos de tempo real do hardware do PC-alvo. Com base na taxa de amostragem selecionada, a aplicação calcula todas as saídas dos blocos a partir de seu modelo.

5. Adquirir sinais - Adquirir sinais de dados utilizando xPC Target Scope. O xPC Target suporta 10 Scopes do tipo alvo, 8 Scopes do tipo arquivo, e um número infinito de Scopes do tipo host, enquanto os recursos do PC alvo possam apoiá-los.
6. Modificar por exemplo, entradas de referência, taxas de amostragens, de acordo com as necessidades de controle.

3.7 Comunicação por Rede

3.7.1 Vantagens da Comunicação em Rede

A comunicação host-alvo utilizando protocolo TCP/IP possui algumas vantagens em relação a comunicação serial:

- **Maior velocidade** - Comunicação de rede utilizando Ethernet pode transferir informações a 10 megabits/segundo, 100 megabits/segundo, ou 1 gigabit/segundo.
- **Maior distância entre PC host e PC alvo** - Com o uso de repetidores e gateways é possível uma maior distância entre host e alvo. Com a comunicação serial isto não é possível.

3.7.2 Placas de Rede

Para a comunicação por rede, é necessário que o PC-alvo e o PC-host possuam placas de redes compatíveis com o produto xPC Target. Essas placas são listadas na Tabela 1. Além das placas apresentadas acima, o xPC Target também é compatível com algumas outras placas de rede que são listadas na Tabela 2.

Tabela 3.1: Placas de rede compatíveis com o produto xPC Target

Board Type	Ethernet Chips Sets	Specific Chips	Select TCP/IP Target Driver Option
ISA, PCI	Novell Ethernet 2000 compatible family	NE2000	NE2000
ISA	SMC LAN91C9X compatible	SMC91C9X	SMC91C9X
PCI	Intel 8255X	Intel 8255X family, including I82559, I82559ER, I82562EM, I82562, I82551, I82551ER, I82562EZ and I82550	I82559
PCI	AMD79C971 PCNET	Fast Ethernet controller chips and other chips in the AMD 79C97x family.	RTLANCE
PCI	Realtek RTL8139, RTL8139C, RTL8139C+	The MathWorks has tested and verified	R8139
	RTL8139C, RTL8139C+	Zonet ZEN3200 and	
	RTL8139C, RTL8139C+	AOpen AON-325.	
PCI	3Com 3C90x	Excludes 3C905-TX board and 3C900-COMBO Etherlink XL	3C90x
PCI	National Semiconductor DP83815	NS83815	NS83815
PCI	Intel I8254X	Fast Gigabit Ethernet controller chips and other chips in the Intel 8254X family.	I8254x

3.8 Propriedades para Comunicação TCP/IP

O ambiente xPC Target deve ser configurado corretamente antes de iniciar. As propriedades que devem ser configuradas informam o xPC Target sobre o software e o hardware que serão utilizados para estabelecer a comunicação. Depois de instalado, deve-se configurar as propriedades do xPC Target para o PC-alvo e para o PC-host como segue:

1. Abra o xPC Target Explorer digitando:

```
xpcexplr
```

A janela xPC Target Explorer abrirá.

xPC Target Explorer irá associar as propriedades de comunicação com o PC alvo.

2. No xPC Target Explorer, expanda o item **TargetPC**.

Opções para configurações irão aparecer em Configuration. Abaixo, têm-se as opções de comunicação, configuração e aparências em, respectivamente, **Communication**, **Settings**, and **Appearance**.

3. Selecione **Communication**.

As configurações para comunicação irão aparecer do lado direito

4. Para **Host target communication**, selecione **TcpIp**.

Deve-se configurar corretamente as propriedades de comunicação de rede que são mencionadas abaixo.

TcpIp target port - Esta propriedade é configurada com **22222**. Este valor não causará problemas.

TcpIp gateway address - Esta propriedade é configurada com **255.255.255.255**. Se o computador estiver conectado com um cabo de rede tipo par trançado, não modifique este valor.

TcpIp target driver - Selecione NE2000, SMC91C9X, I82559, RTLANCE, R8139, 3C90x, NS83815, ou I8254x de acordo com sua placa de rede.

TcpIp target bus type - Escolha **ISA** ou **PCI** de acordo com a placa de rede.

Tabela 3.2: Placas de rede compatíveis com o produto xPC Target

Board Type	Board Number	xPC Target Driver
PCI	SMC EZ Card 10 SMC1208T (RJ45)	NE2000
PCI	SMC EZ Card 10 SMC1208BT (RJ45, BNC)	NE2000
PCI	SMC EZ Card 10 SMC1208BTA (RJ45, BNC, AUI)	NE2000
PCI	Intel PRO/100 S	I82559
PCI	Intel PRO/100 M	I82559
PCI	Zonet ZEN3200	R8139
PCI	AOpen AON-325	R8139
PCI	Boards compatible with 3C90x, except the 3Com 3C905-TX Fast EtherLink XL PCI board	3C90x
PCI	Argon 10/100MB National Semiconductor DP83815 PCI Fast Ethernet Adapter	NS83815
ISA	SMC EZ Card 10 SMC1660T (RJ45)	NE2000
ISA	SMC EZ Card 10 SMC1660BT (RJ45, BNC)	NE2000
ISA	SMC EZ Card 10 SMC1660BTA (RJ45, BNC, AUI)	NE2000
PC/104	RealTime Devices USA CM202 (RJ45, BNC, AUI)	NE2000
PC/104	WinSystems Inc. PCM-NE2000-16 (RJ45)	NE2000
PC/104	WinSystems Inc. PCM-NE2000-16-BNC(BNC)	NE2000
SBC	Versallogic VSBC-6	SMC91C9X

3.9 Opções do xPC Target Boot

Pode-se iniciar o PC alvo com o xPC Target *kernel* utilizando um dos caminhos abaixo a partir do xPC Target Explorer em que:

CD Boot - Inicia o *kernel* no PC alvo com um CD.

Boot Floppy - Inicia o *kernel* no PC alvo com um disquete 3.5-inch.

Network Boot - Inicia o *kernel* no PC alvo através da comunicação TCP/IP.

Neste trabalho foi utilizado disquete de 3.5-inch, o qual será criado para carregar e rodar o xPC Target kernel no PC-alvo. Depois de mudar as propriedades de comu-

nicação, será necessário criar ou atualizar o disquete. Para criar o disquete com as configuração desejadas, seguir as seguintes instruções:

1. Na janela do **xPC Target Explorer** no **xPC Target Hierarchy**, selecione **Configuration** em **TargetPC1**.
2. Selecione **Boot Floppy** como mostrado na Figura 3.4.
3. Se necessário modifique o drive **a:**.
4. Click **Apply**.
5. Click **Create Bootdisk**.
6. Insira o disquete no drive e click **OK**.
7. Quando o drive de disco parar, remova o disquete.
8. Insira o disquete no PC alvo e reinicie o PC alvo.

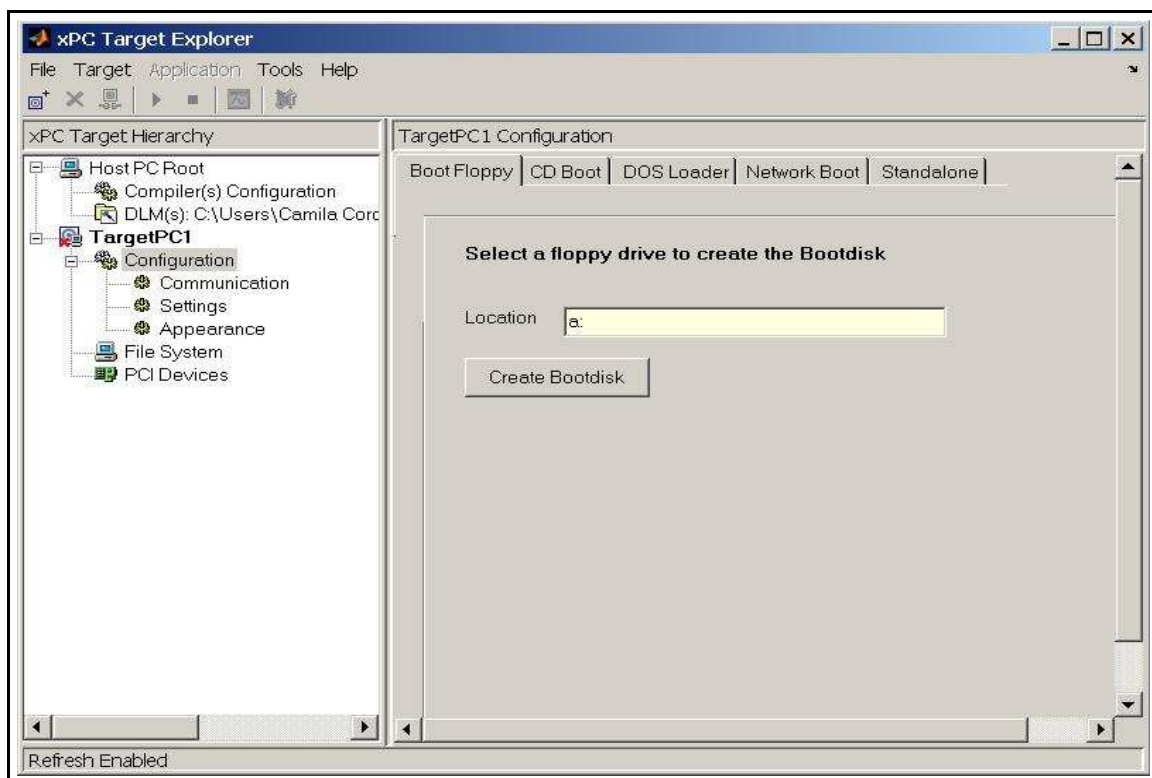


Figura 3.4: Gravando disco com o kernel.

3.10 Iniciando o xPC Target

3.10.1 Iniciando o PC-alvo

Quando iniciado, o PC-alvo carrega, inicia o xPC Target kernel e aguarda o PC-host até o envio de uma aplicação. Após todas configurações serem realizadas é necessário executar o comando **Build**, o qual carregará automaticamente a aplicação no PC-alvo. Os passos abaixo descrevem como realizar este processo.

1. Insira o disco, criado no PC-host, no PC-alvo.
2. Ligue ou reinicie o PC-alvo.

Com isto, no PC-alvo, será mostrada uma tela como na Figura 3.5, onde é possível ver que o kernel foi carregado e que está aguardando para carregar uma aplicação. 1 MB de memória é reservado para a aplicação, 3MB é utilizado pelo kernel e 28 MB é livre de um total de 32 MB. O xPC Target kernel utiliza 28 MB para a pilha, saídas gráficas e aquisição de dados.

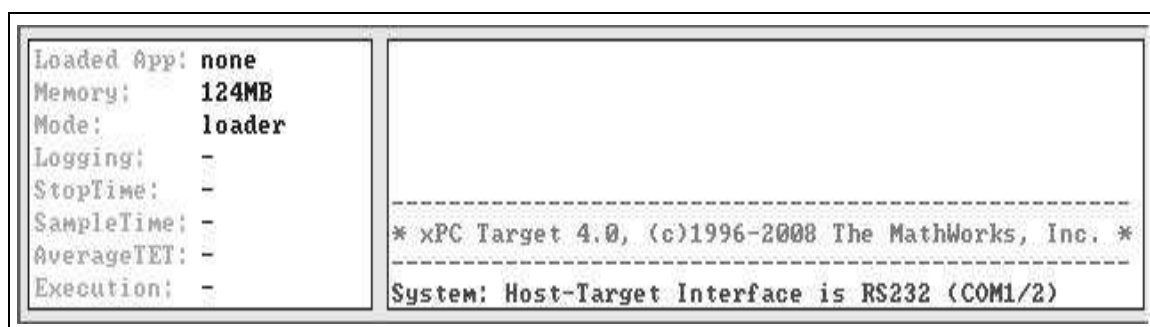


Figura 3.5: *xPC Target no PC-alvo.*

A próxima tarefa é configurar os parâmetros de tempo real do Real-Time Workshop.

3.10.2 Configurando Parâmetros de Tempo-real

Alguns parâmetros de tempo real devem ser configurados para fornecerem informações para o Real-Time Workshop em como criar uma aplicação alvo no Simulink.

1. Abra sua aplicação xPC Target.

2. Na janela do **Simulink**, em **Simulation menu**, click em **Configuration Parameters**.

Uma janela como na Figura 3.6 aparecerá.

3. Click em **Real-Time Workshop**.
4. Para criar uma aplicação, na seção **Target selection**, click em **Browse** para abrir a lista para o *System target file*. Escolha **xpctarget.tlc**, e click **OK**.

Com esta opção (xpctarget.tlc) é configurado automaticamente o template makefile (xpc_default_tmf) e o make command (make_rtw) como mostrado na Figura 3.6

5. Na parte esquerda, click em **xPC Target options**.
6. Para **Execution mode**, escolha **Real-Time**.
7. Para **Real-time interrupt source**, selecione **Timer**.
8. Selecione **Log Task Execution Time**.
9. Em **Name of xPC Target object created by build process**, entre como o nome do objeto que será criado após o comando *Build*.

O janela deve fica configurada como a Figura 3.7.

10. Click em **Ok**.

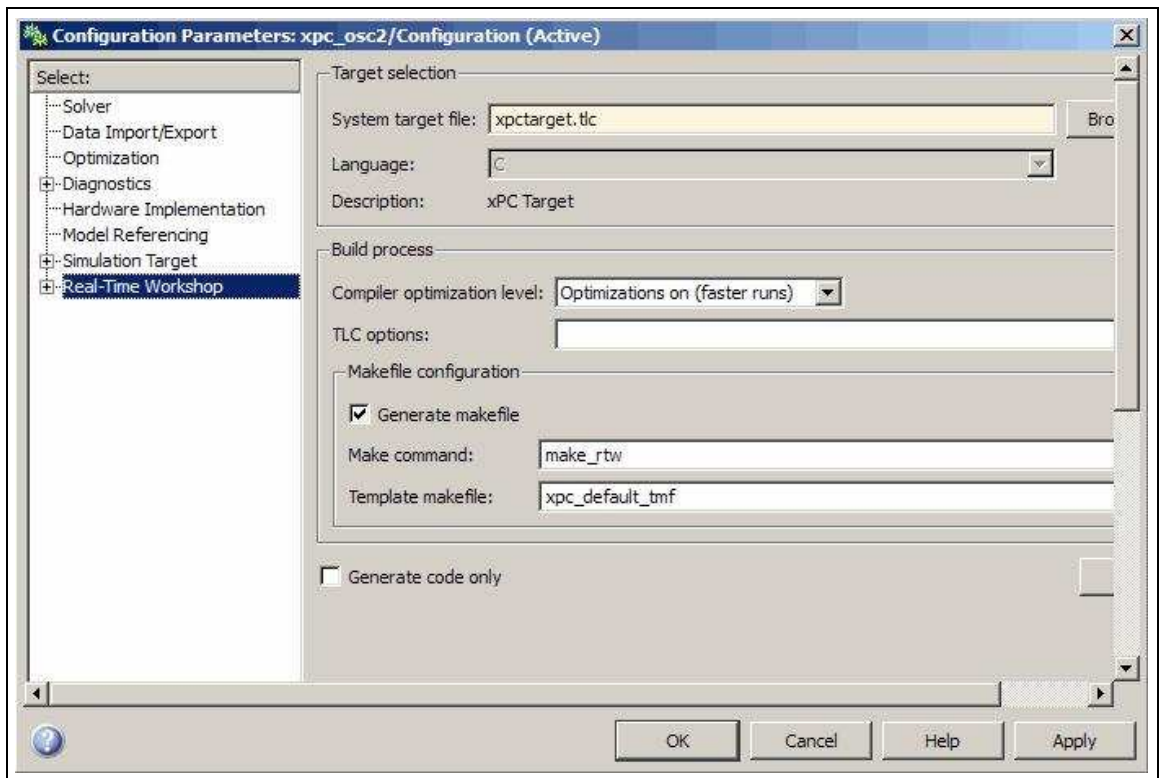


Figura 3.6: Configuração de parâmetros de tempo real: Real-Time Workshop.

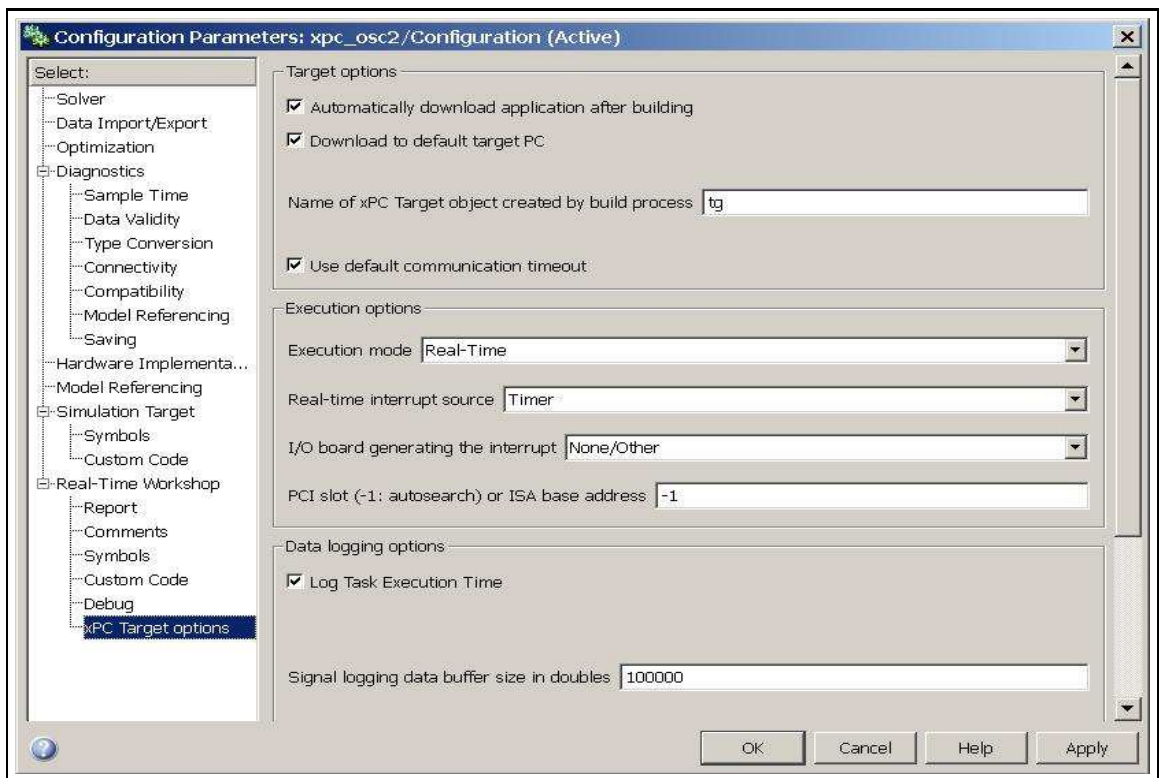
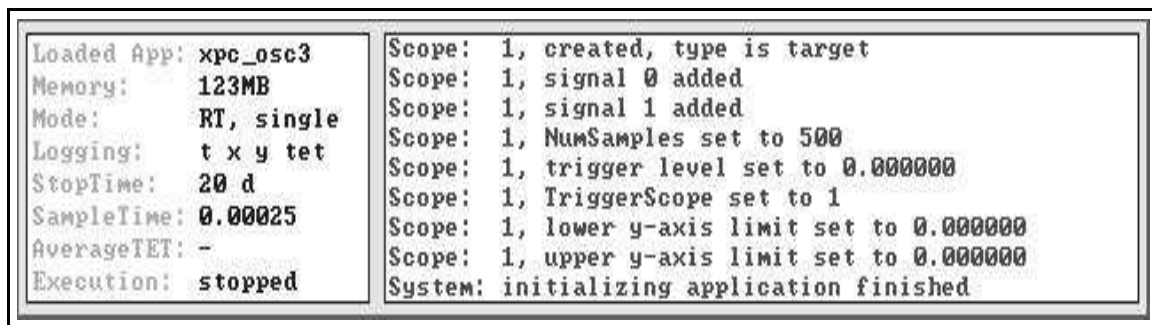


Figura 3.7: Configuração de parâmetros de tempo real: xPC Target options.

3.10.3 Carregando a Aplicação Alvo

O comando **build** gera, compila e carrega o código C, que é criado a partir de sua aplicação, no PC-alvo. Para isto, depois de feitas todas configurações mencionadas anteriormente, na janela do Simulink, em **Tolls**, selecione **Real-Time Workshop**. Dentre as opções em Real-Time Workshop, escolha **Build Model**. Como ilustração temos a Figura 3.8 onde uma aplicação como o nome de `xpc_osc3` foi carregada.



```
Loaded App: xpc_osc3      Scope: 1, created, type is target
Memory:      123MB       Scope: 1, signal 0 added
Mode:       RT, single   Scope: 1, signal 1 added
Logging:    t x y tet    Scope: 1, NumSamples set to 500
StopTime:   20 d        Scope: 1, trigger level set to 0.000000
SampleTime: 0.00025     Scope: 1, TriggerScope set to 1
AverageTET: -          Scope: 1, lower y-axis limit set to 0.000000
Execution:  stopped     Scope: 1, upper y-axis limit set to 0.000000
                    System: initializing application finished
```

Figura 3.8: *PC-alvo carregado com a aplicação xpc_osc3.*

3.10.4 Iniciando o xPC Target

Após carregar a aplicação alvo basta clicar em **Connect to Target button** para conectar o PC-host ao PC-alvo e então iniciar a aplicação no botão **Start Application button**. Ambos se encontram na janela do xPC Target Explorer. Após iniciado, a janela do xPC Target Explorer deve ter a aparência da Figura 3.9.



Figura 3.9: *Interface xPC Target no PC-alvo.*

Capítulo 4

Aplicações Realizadas

4.1 Aplicação Real-Time Workshop

Como primeira aplicação do Real-Time Workshop foi realizado o envio e a recepção de dados como ilustrado na Figura 4.1. Sinais, enviados por um gerador de sinais, puderam ser visualizados em um Scope (Figuras 4.2, 4.3, 4.4 e 4.5) e outros, enviado pelo Simulink, visualizados em um osciloscópio (Figuras 4.6 e 4.7). Em seguida foi realizado o controle em malha fechada de um motor de corrente contínua (CC) Eletrocraft de 60V com corrente máxima de 5A, velocidade máxima 6000rpm e torque máximo 0.353 Nm com tacogerador acoplado. A função de transferência do motor CC utilizado é dada por

$$G(s) = \frac{442040}{s^2 + 705s + 42307}. \quad (4.1)$$

Para o controle do motor CC foi utilizado um controlador P. O diagrama do Simulink para este processo é mostrada na Figura 4.8. Um módulo PWM foi utilizado para gerar um sinal do tipo dente-de-serra com uma amplitude constante igual a 10V e frequência ajustável entre 2KHZ e 20KHZ. Este sinal que é comparado com um sinal CC, entregue pelo controlador P do Simulink e fornecido por uma placa NI AT-MIO-16E-10. O resultado dessa comparação, que são os pulsos modulados em largura e período fixo, é fornecido como sinal de saída do PWM. Como este sinal não é capaz de fornecer um nível de potência suficiente para acionar o motor CC, faz-se necessário a implementação de um circuito adicional que possa elevar o nível de potência dos pulsos do módulo PWM e acionar o motor CC. Este circuito é formado por um transistor de potência

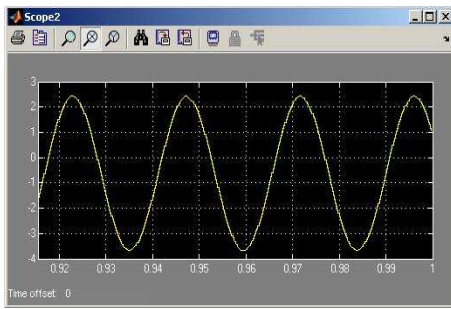


Figura 4.2: Senóide de 40 Hz.

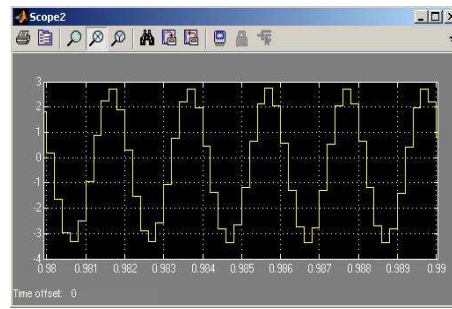


Figura 4.3: Senóide de 200 Hz.

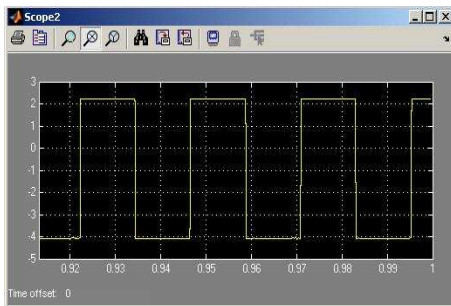


Figura 4.4: Onda quadrada de 40 Hz.

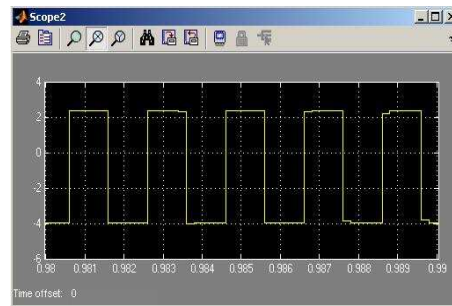


Figura 4.5: Onda quadrada de 200 Hz.

(TIP 120) ligado a uma fonte externa para operar em modo de chaveamento de forma a fornecer níveis de corrente necessários para acionar o motor CC.

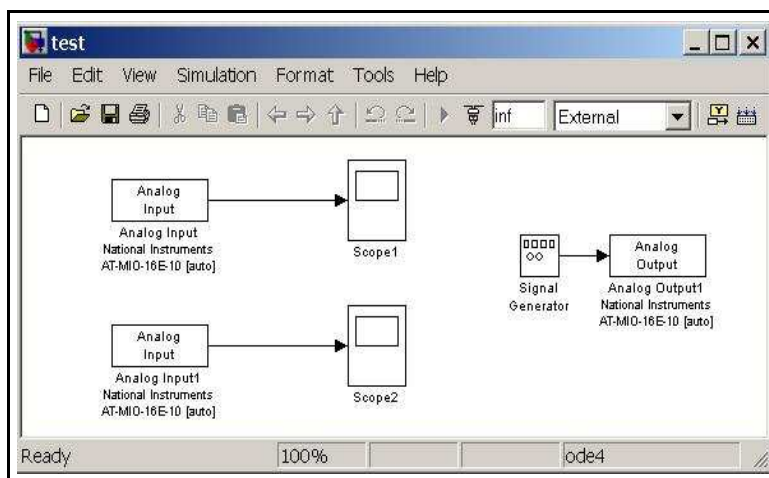


Figura 4.1: Entrada e saída analógica.

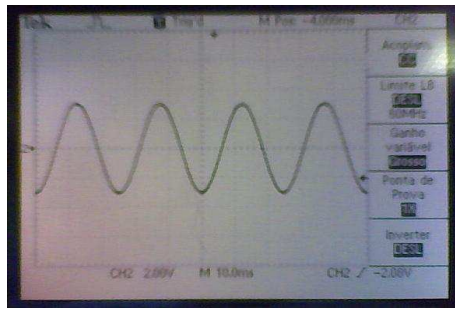


Figura 4.6: Senoide de 40 Hz.

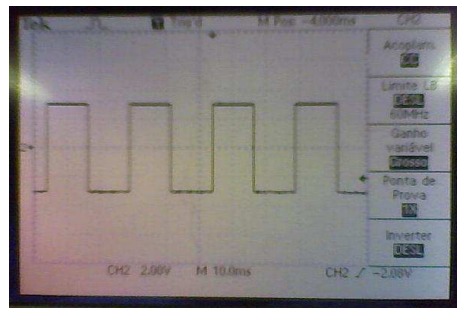


Figura 4.7: Onda quadrada de 40 Hz.

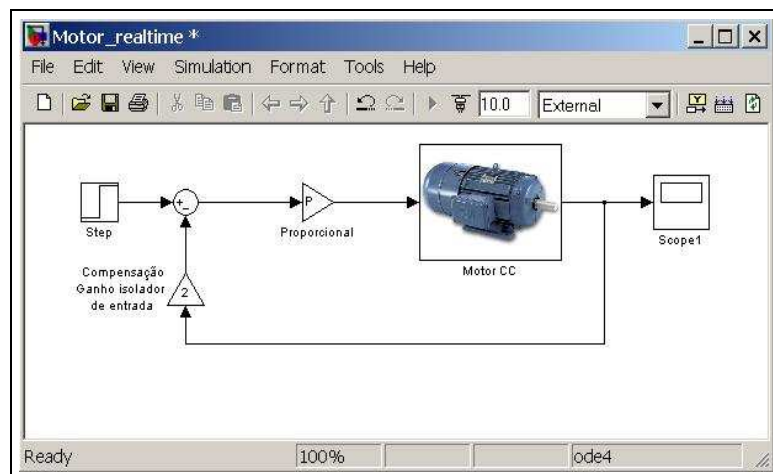


Figura 4.8: Controle de um motor CC através do simulink.

Para a entrada e saída de sinal analógico é necessário adicionar blocos para estas funções. Pode-se conferi-los abrindo o bloco com a figura do motor CC da Figura 4.8. A Figura 4.9 ilustra estes blocos. Assim, basta alimentar o motor com uma tensão de referência através do transistor de potência e fechar a malha de controle com o sinal proveniente do taco gerador do motor. O esquemático do circuito empregado e do motor CC está ilustrado na Figura 4.10.

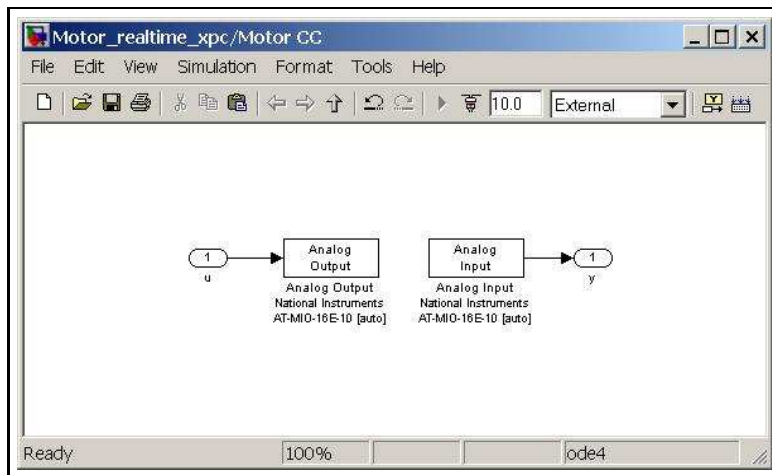


Figura 4.9: Blocos de entrada e saída analógica.

Após realizado o controle do processo real, pode-se constatar a eficiência da ferramenta através da saída gráfica de tensão do taco gerador (Figura 4.11) para uma referência degrau. Observa-se que a velocidade do motor estabilizou.

Nesta etapa do projeto observa-se um importante resultado. A placa de aquisição de dados *NI AT-MIO-16E-10* vinha sendo utilizada apenas com o aplicativo *Lab View*. Este aplicativo limita o período de amostragem a 1 milissegundo. Com este valor de amostragem, pode-se ocorrer algumas limitações no monitoramento de variáveis ao realizar o controle de um robô móvel, por exemplo. Com a implementação em Matlab e Simulink consegue-se chegar a um período de amostragem de até 10 microssegundos. Assim, com a utilização do controle em tempo real via Matlab e Simulink, plantas com dinâmicas mais rápidas podem ser controladas com a mesma placa de aquisição.

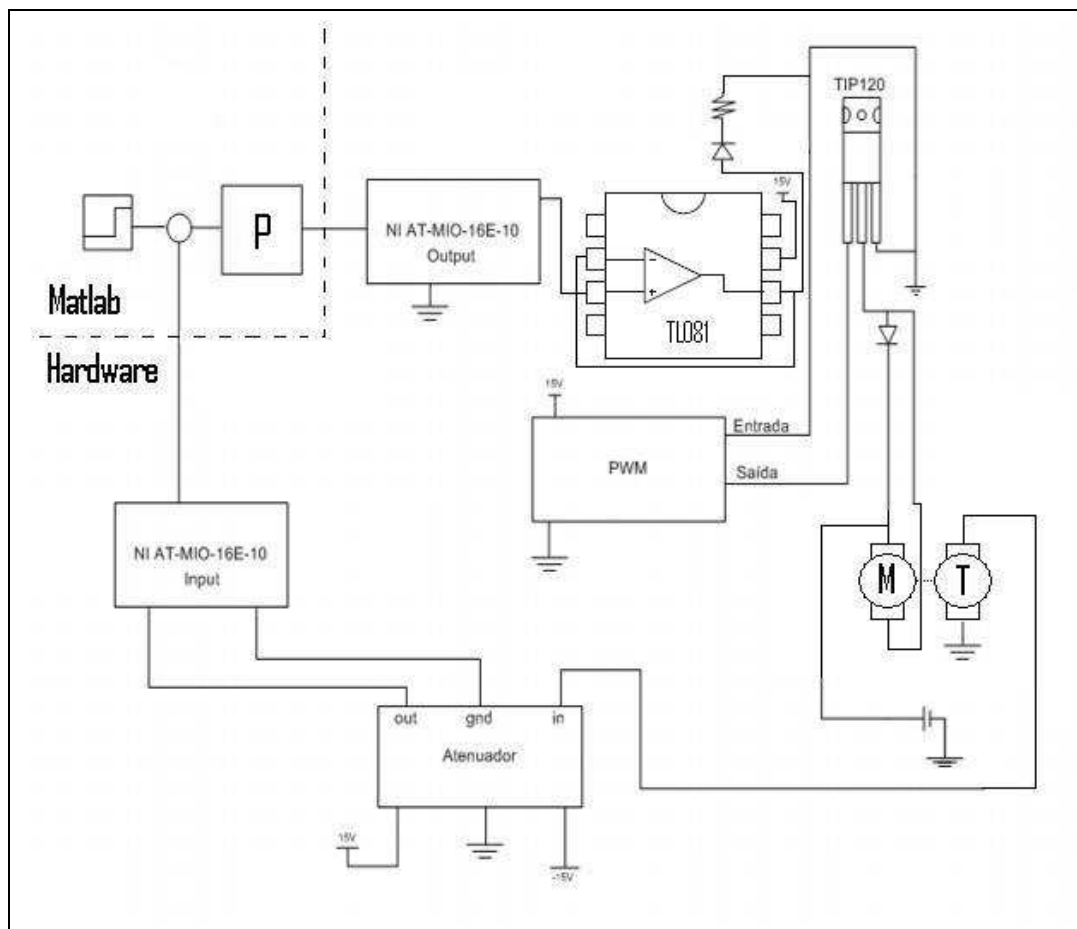


Figura 4.10: Diagrama esquemático do sistema de controle do motor CC.

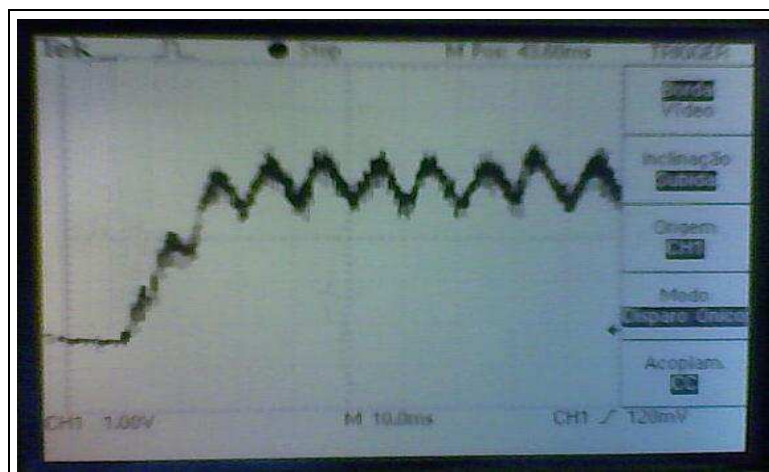


Figura 4.11: Tensão do tacho gerador do motor CC.

4.2 Aplicação xPC Target

Inicialmente foi proposto o controle de um motor de corrente contínua utilizando a ferramenta xPC Target como realizado com o RTW. Porém não foi possível devido a falta de uma placa de aquisição de dados que fosse reconhecida pela ferramenta xPC Target. Devido a isto, foi feita a simulação desse motor e foi enviado ao PC-alvo, através do xPC Target, um gráfico com dados de simulação, podendo assim comprovar que o método RTW realizado anteriormente gerou dados corretos. O sistema pode ser visualizado na Figura 4.12.

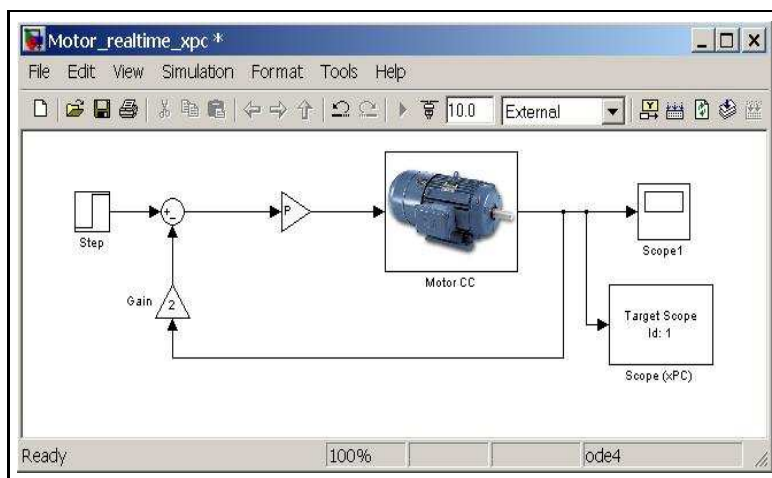


Figura 4.12: Simulação de um motor CC através do xPC Target.

O bloco *Target Scope* envia, de forma gráfica para o PC-alvo, o sinal que está recebendo. Com isto torna-se possível a visualização do processo no PC-alvo também. A Figura 4.13 ilustra a tela visualizada no PC alvo. Esta é a velocidade do motor CC e o degrau aplicado no sistema (5V). Nota-se que a velocidade estabilizou do mesmo modo que a aplicação em tempo real realizada anteriormente a menos do ruído. Os gráficos estão em escalas diferentes pois foi muito difícil conseguir a Figura 4.13 devido ao tempo de resposta do motor ser muito baixo e para conseguir visualizá-lo foi necessário que o simulador parasse alguns milissegundos após o início de sua execução. Devido a isto, uma segunda simulação foi realizada, acrescentando ruído. Para representar o ruído, um sinal senoidal de frequência 2953 rad/s e amplitude se 2.7 vezes a constante do taco gerador ($ktg = 0.15$) foi adicionado na saída do taco gerador juntamente com um sinal aleatório de variância 0.1. Esta simulação pode ser visualizada na Figura 4.14 e pode ser notada sua semelhança com a Figura 4.11 que representa a planta real.



Figura 4.13: *Velocidade de um motor CC visualizada no PC-alvo.*

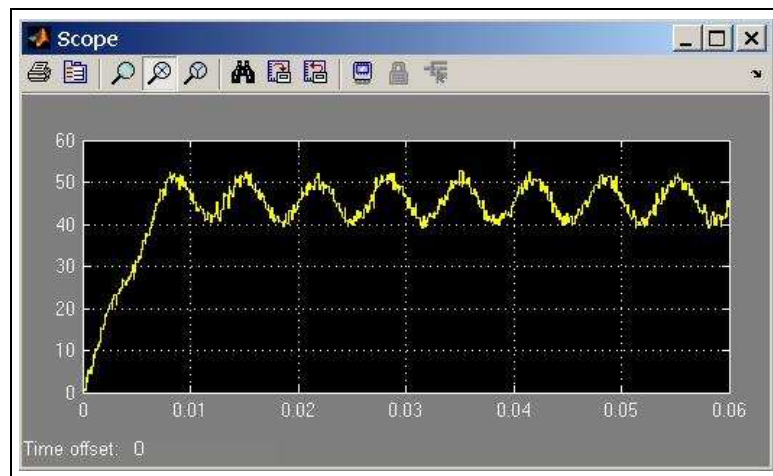


Figura 4.14: *Simulação de um motor CC com ruído.*

Capítulo 5

Conclusão

Neste projeto foram estudadas a implementação de técnicas de controle em plantas reais utilizando a plataforma de desenvolvimento Real-Time Workshop e xPC Target. Para isto, foi utilizado um kernel, o qual torna possível o controle em tempo real através do Matlab e Simulink pois dá prioridade a aplicação em relação aos outros processos do PC. Foi apresentado com detalhes a configuração necessária para a implementação de controle em tempo real. Para testar a configuração de controle, utilizou-se um motor CC com controlador P. Foi constatado que a saída da planta real foi compatível com a simulada e apresentada em um PC-alvo através da ferramenta xPC Target, a qual permite o controle e monitoração de plantas reais a distância. Com essas duas ferramentas é possível realizar o controle de diversas plantas, inclusive a distância.

Referências Bibliográficas

Eker, J. e A.Cervin (1999). A Matlab toolbox for real-time and control systems co-design, *Sixth International Conference on Real-Time Computing Systems and Applications*, Hong Kong, P.R. China, pp. 320–327.

Gómez, O. M. (2004). *Kernel de tiempo real para el control de procesos*, Master's thesis, Centro de Investigación y de Estudios Avanzados del IPN, Departamento de Ingeniería Eléctrica Sección Computación.

Real-Time Workshop Getting Started (2008). The Mathworks Inc., Natick, MA.

Teng, F. C. (2000). Real-time control using MATLAB SIMULINK, *International Conference on Systems, Man, Cybernetics*, Vol. 4, Nashville, TN, USA, pp. 2697–2702.

xPC Target Getting Started (2008). The Mathworks Inc., Natick, MA.