

Laboratório 5

SAÍDAS GRÁFICAS E PARA ARQUIVO NÃO TEMPORIZADAS

5.1 Objetivos

Aprofundar o estudo e investigações dos recursos de programação de modos estruturados não temporizados e implementar saídas gráficas em painéis e para arquivo.

5.2 Generalidades.

Na aula anterior foram estudados os recursos de estruturação de programas através de *Loops* e sequenciamento de rotinas e ou tarefas. Estes recursos têm como finalidade a organização de tarefas em um programa e principalmente indexar as etapas e variáveis envolvidas em cada tarefa.

É muito comum nestas ocasiões o interesse ou mesmo a necessidade de se indicar os resultados sequenciais em um tipo de forma gráfica ou em arquivos para que se possa investigar o andamento ou re-processar os dados *Off-Line*.

É importante lembrar da aula anterior que toda variável indexada em um *Loop* dever ser habilitada para indexação na saída dos *Loops* e são inerentemente variáveis tipo *Array*, ou seja, são vetores linhas ou colunas. Portanto os recursos de manipulação de *Arrays* são importantes.

Para as atividades específicas do curso de controle digital, o importante será ter um razoável domínio dos recursos gráficos em painel e o registro de dados em arquivos que possam ser transferidos para outros pacotes de software tais como o Matlab.

5.3 Saídas gráficas

O LabVIEW dispõe de basicamente três tipos de saídas gráficas que são disponíveis na janela Painel e no *Pop-Menu* CONTROLS - *Graph*. Estas funções são *Chart*, *Graph* e *XY-Graph*. Cada uma destas opções pode servir a diferentes propósitos como será apresentado a seguir e posteriormente treinado em atividades no laboratório.

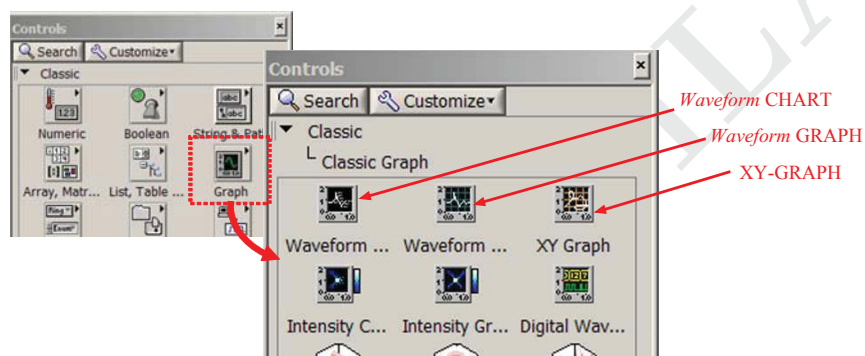


Figura 5.1: Menu e sub-menu de recursos gráficos

Toda saída gráfica alocada por meio da janela Painel irá alocar um correspondente elemento na janela de Diagrama, o qual deverá ser devidamente conectado na finalização do VI.

5.3.1 Saída *Chart*

As saídas do tipo *Chart* são utilizadas para as tarefas mais simples de expressão gráfica de resultados. O *Chart* não necessita da grandeza no eixo das ordenadas e ainda pode indicar valores escalares em forma gráfica. Como entrada o *Chart* aceita valores escalares, vetores de uma dimensão e vários vetores de uma dimensão e ainda entradas tipo WDT (*Waveform-Data-Type*): que contém informações de data-hora). As saídas de *Chart* são tipo *Appended*, ou seja, cada nova variável ou *Array* passado para o *Chart* é acrescido na tela até que se exceda o tamanho do *Buffer* de entrada que na realidade é o tamanho da variável *Chart*.

As várias opções e recursos de representação gráfica podem ser acessados na janela PAINEL acionando-se o Pop-Menu do elemento *Chart*. As principais destas opções no

Pop-Menu estão resumidas na Opção Propriedades e são:

Opção *Show* : onde se pode habilitar ou não a indicação das escalas XY, da legenda, *Label*, *Zoom*, cores e estilos dos gráficos e reticulado, etc.



Figura 5.2: Exemplo de saída tipo *Chart*

Opção *Data Operation (Clear Chart)*: onde se pode reinicializar o *Chart*, Limpar toda tela e estabelecer os modos auto-escala XY.

Opção *Chart History length* : onde se pode estabelecer o tamanho da variável do *Chart* que será preenchida.

Opção *Overlay-Stack Plots* : esta opção tem função quando as variáveis são passadas para o *Chart* em forma de Agrupamento ou *Cluster*. Desta forma é possível colocar vários gráficos (de várias grandezas) no mesmo *Chart* ou em *sub-plots* (em *displays* gráficos separados).

Este modo pode ser gerado de duas maneiras diferentes tal como ilustrado na figura 5.3 usando-se o recurso *Build-Array* ou *Bundle*. O resultado deste tipo de apresentação gráfica é indicado na figura 5.4.

Procure inspecionar cada uma das propriedades acessando-se o item correspondente de propriedades listadas no Pop-Menu.

5.3.2 Saída tipo *Graph*

Este tipo de saída permite se conectar uma variável como ordenada (eixo x) no gráfico de saída além de permitir múltiplos gráficos com a diferença de que neste caso a entrada deve sempre ser do tipo *Array* ou do tipo *WDT*. As opções de visualização e configuração são praticamente as mesmas do *Chart* e também acionadas via Pop-Menu.

Uma das maneiras de se usar o *Graph* é simplesmente conectar a variável ou variáveis colecionadas em *Array* ao elemento *Graph*. A principal diferença com relação ao *Chart* é

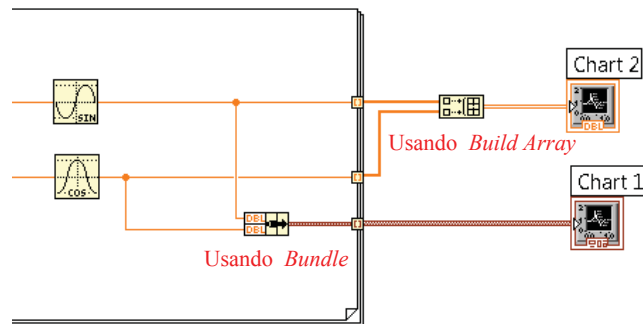
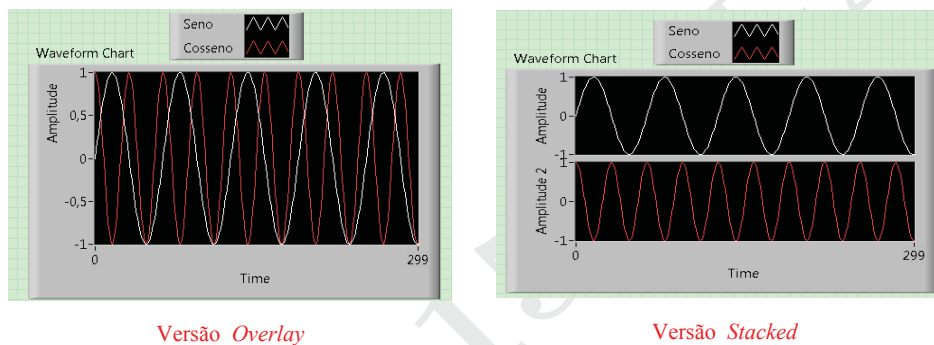


Figura 5.3: Recursos de agrupamento de dados

Figura 5.4: Visualização dos recursos *Overlay* e *Stacked* do *Chart*

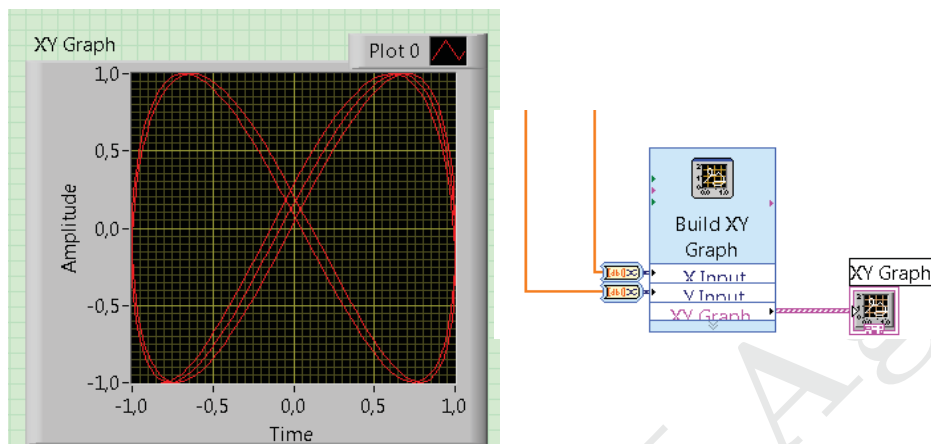
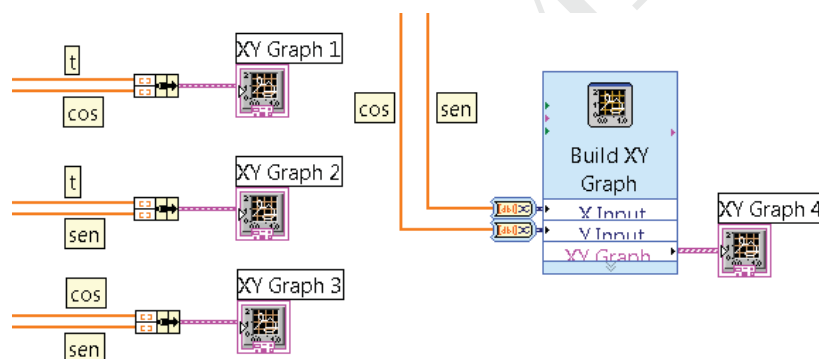
que o eixo das ordenadas (eixo y) será simplesmente o número de indexação da variável, ou seja do tamanho exato da dimensão do *Array* independente se se executar uma única vez ou executar continuamente. Desta forma o *Graph* se comporta tal como o *Chart*. Um exemplo de saída tipo *Graph* é visto na figura 5.5

Para se obter múltiplos gráficos no mesmo *display*, usando ou não o arranjo do eixo x, colecciona-se inicialmente os pontos em um *Array* múltiplo e conecta-se ao *Graph*.

5.4 Saída tipo *Graph XY*

Este tipo é semelhante ao do *Graph*, mas permite conectar diferentes variáveis respectivamente aos eixos X e Y. Para tal, as variáveis devem ser coletadas com auxílio de um Agrupamento tal como na figura 5.6. Quando o elemento de saída *Graph-XY* é obtido a partir da janela *Controls/Classic* usa-se os elementos *Bundle* para agrupar os sinais a serem plotados.

Na versão do LabVIEW 2013, a inserção do elemento *Graph-XY* a partir da janela

Figura 5.5: Exemplo de saída *Graph*Figura 5.6: Exemplo de agrupamentos para o uso do *Graph-XY*

Control/Express já contém um módulo *Build XY Graph* na janela *DIAGRAMA* e que facilita a montagem do vetor de variáveis a serem manipuladas no gráfico XY. Um exemplo disto é visto na figura 5.7.

5.5 Manipulação de dados de arquivos ou para arquivos

Este tipo de operação pode ser de interesse para se fazer registro periódico ou contínuo de grandezas que determinam o funcionamento de um VI e possibilita a armazenagem ou leitura de dados em arquivos. Este recurso será usado para armazenar em arquivo todos os dados de interesse do experimento e depois carregá-los no ambiente Matlab e

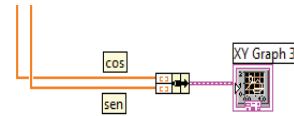
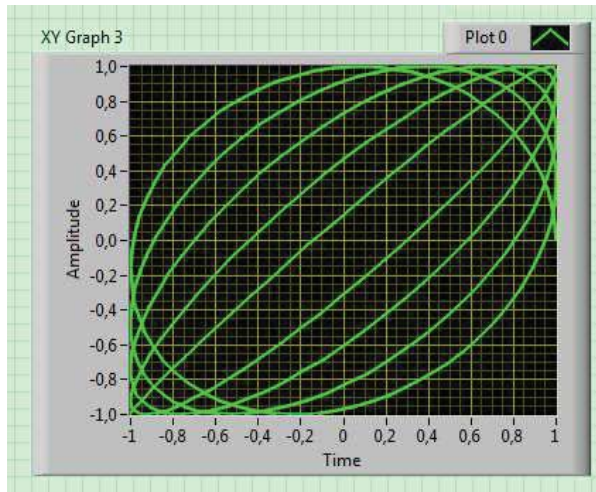


Figura 5.7: Exemplo de uso do *Graph-XY*

comparar com as simulações do Simulink.

Com relação à disciplina Controle Digital, estamos interessados em fazer o registro das entradas, saídas, e a das ações de controle nos experimentos a serem estudados.

O acesso às funções de arquivo é feito na janela DIAGRAMA no Pop-Menu *Programming/File I/O* e contém as opções tais como indicado na figura 5.8.

As duas primeiras opções permitem escrever e ler dados numéricos em um arquivo enquanto que a terceira e quarta permitem manipular apenas caracteres (*strings*). As demais opções são de caráter específico e não serão abordadas aqui.

Para se armazenar um dado em um arquivo, escolhe-se o elemento *Write to Spreadsheet File* e provê-se a devida conexão do mesmo. As conexões principais deste elemento são:

Path File : onde se insere o nome do arquivo e o diretório do mesmo;

1D ou 2D data : entrada para a variável *Arrays* tipo 1D ou 2D.

Append to File : anexa novos dados em arquivos existentes ou *True* abre novo arquivo.

Transpose : transpõe os dados 1D ou 2D antes de armazenar, ou seja, arquiva em colunas.

As demais conexões são opcionais. O **Path File** pode ser uma constante ou uma entrada de controle a ser editada no painel. Deve-se lembrar que este elemento converte os números em equivalentes caracteres ASCII antes de armazenar. Nesta operação deve-se verificar forma de indicação do ponto decimal (ponto ou vírgula) antes de se carregar estes dados em outro programa. O Matlab, por exemplo só entende a separação decimal

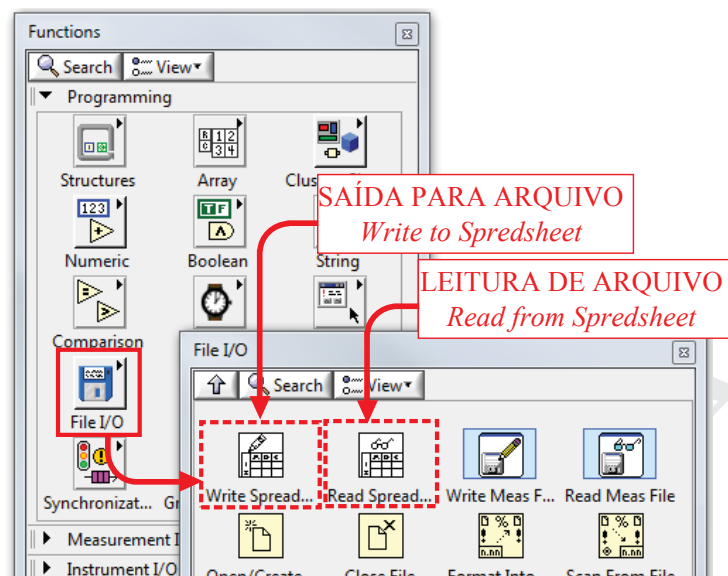
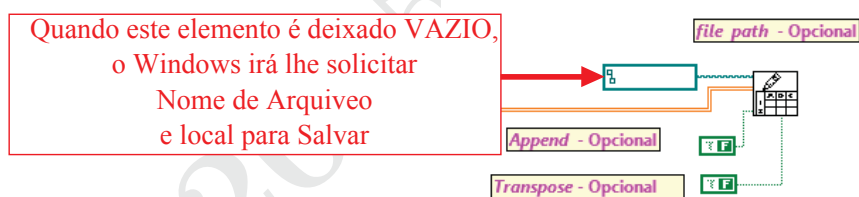


Figura 5.8: Funções de Entrada e Saída em Arquivos

Figura 5.9: Exemplo de Saída de Dados com *Write-to-Spreadsheet*

com ponto.

5.6 Interagindo com o Matlab por meio do LabVIEW : MATLAB-Script

Este recurso, *MATLAB-Script*, permite que se executem comandos independentes do Matlab ou ainda rotinas completas editadas com a sintaxe do Matlab e armazenadas em arquivos com extensão *.m*. Este recurso faz parte das ferramentas de Fórmulas de Análise Matemática, tal como o *Formula-Node* vista na aula anterior.

A estrutura *MATLAB-Script* pode apenas receber um dado do LabVIEW e executar um comando, pode apenas gerar um dado para o LabVIEW ou pode receber um dado do

5.6. INTERAGINDO COM O MATLAB POR MEIO DO LABVIEW : MATLAB-SCRIPT39

LabVIEW, executar um comando (ou rotina) Matlab e retornar dados para o LabVIEW. Portanto esta estrutura pode ser configurada com apenas uma entrada, ou com apenas uma saída ou com entradas e saídas operadas exatamente como no *Formula-Node*.

Este recurso pode ser encontrado na janela DIAGRAMA via Pop-Menu *Functions - Mathematics - Scripts and Formula* segundo a ilustração da janela DIAGRAMA da figura 5.10.

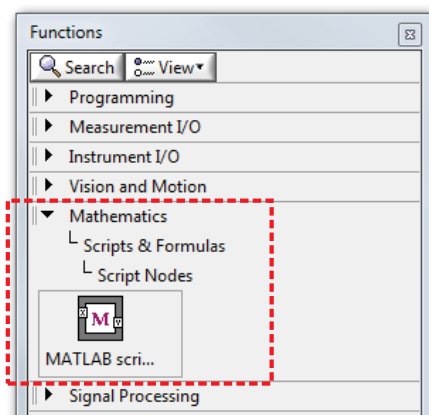


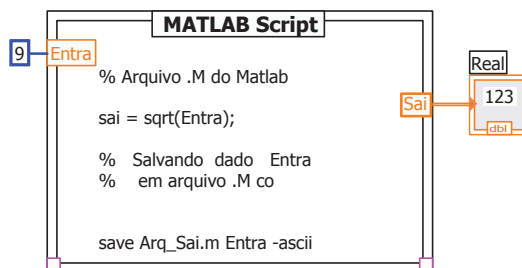
Figura 5.10: Função *Script-Node* do Matlab

Ao se selecionar o elemento *MATLAB-Script* tal com acima e colocá-lo na janela DIAGRAMA surgirá uma janela de *MATLAB-Command-Window* semelhante à janela de comandos do Matlab. O programa Matlab não precisa estar carregado na memória do Micro, mas o micro precisa ter uma instalação do Matlab.

Os dados gerados e manipulados na janela de comando via LabVIEW não interferem com a janela e dados do programa Matlab se o mesmo estiver operando ou for acionado. Quando o Matlab estiver em operação ocorrerão duas janelas de comando e que não interferem entre si.

Para que os dados gerenciados pelo LabVIEW com o *MATLAB-Script* esteja realmente disponível no programa Matlab é preciso salvar os dados de interesse em formato .MAT ou em arquivo ASCII com extensão .M.

No exemplo a seguir, a variável “Entra” gerada no LabVIEW é processada pelo comando raiz-quadrada (`sqrt()`) do Matlab e depois é salva no arquivo `Arq_Sai.m` com o comando `save` do Matlab.

Figura 5.11: Exemplo de *Script-Node* do Matlab

5.7 Atividades de aplicação

Procure executar os VIs com as funções solicitadas a seguir, sem utilizar recursos de temporização que serão vistos na próxima aula. Procure treinar também a geração e gerenciamento do VI por meio de um PAINEL equivalente dotado de entradas, saídas e mensagens *strings*.

- i) - Elabore um PAINEL com um indicador tipo *Chart* e conecte a um controle deslizante escalar. Teste neste exemplo simples as opções de recursos, tais como indicação gráfica (linhas cheias, pontilhadas, pontos simples, cores, reticulado etc).
- ii) - Altere o VI anterior inserindo uma estrutura em *Loop* gerando uma variável *Array* e teste a mesma no indicador *Chart*.
- iii) - Use um elemento de Agrupamento (*Bundle*) dentro da estrutura de *Loop* e *Build-Array* fora do *Loop* e indique os resultados em um *Chart*. Teste as opções de gráfico com as propriedades *Overlay* e *Stack Plots*.
- iv) - Usando alguma estrutura de *Loop*, gere um VI com um indicador *Graph* simples.
- v) - Experimente agregar um segundo *Array* ao indicador do item (iii).
- vi) - Gere um VI com um indicador *Graph* dotado de arranjo do eixo X usando *Bundle* tal como indicado anteriormente. (Use um *Bundle* fora do *Loop*).
- vii) - Gere um VI com duas variáveis do tipo *Array* e use um indicador *Graph-XY* para expressar ambos *Arrays*. (Use um *Bundle* dentro do *Loop*).
- viii) - Use um dos VIs anteriores e teste a armazenagem dos dados em arquivo. Tente abrir este arquivo no Matlab, plote os pontos do arquivo e compare os gráficos do LabVIEW e do Matlab.

Desafio Prática 5

ix) - Executar um VI para gerar 3 ciclos de uma senoide com 20 pontos por ciclo. Indicar a senoide numa janela gráfica de PAINEL

x) - Salvar em um arquivo de dados a senoide do item (ix) . Importar o arquivo no MATLAB, gerar os mesmos 3 ciclos de senoide com 100 pontos no Matlab e comparar graficamente os dois resultados.

USE 'x' OU 'o' PARA INDICAR OS PONTOS IMPORTADOS DO LABVIEW

OBS - OS ITENS (ix) E (x) DEVERÃO CONSTAR NO RELATÓRIO 1.
Indicar VI e procedimento Matlab/Simulink e Resultado