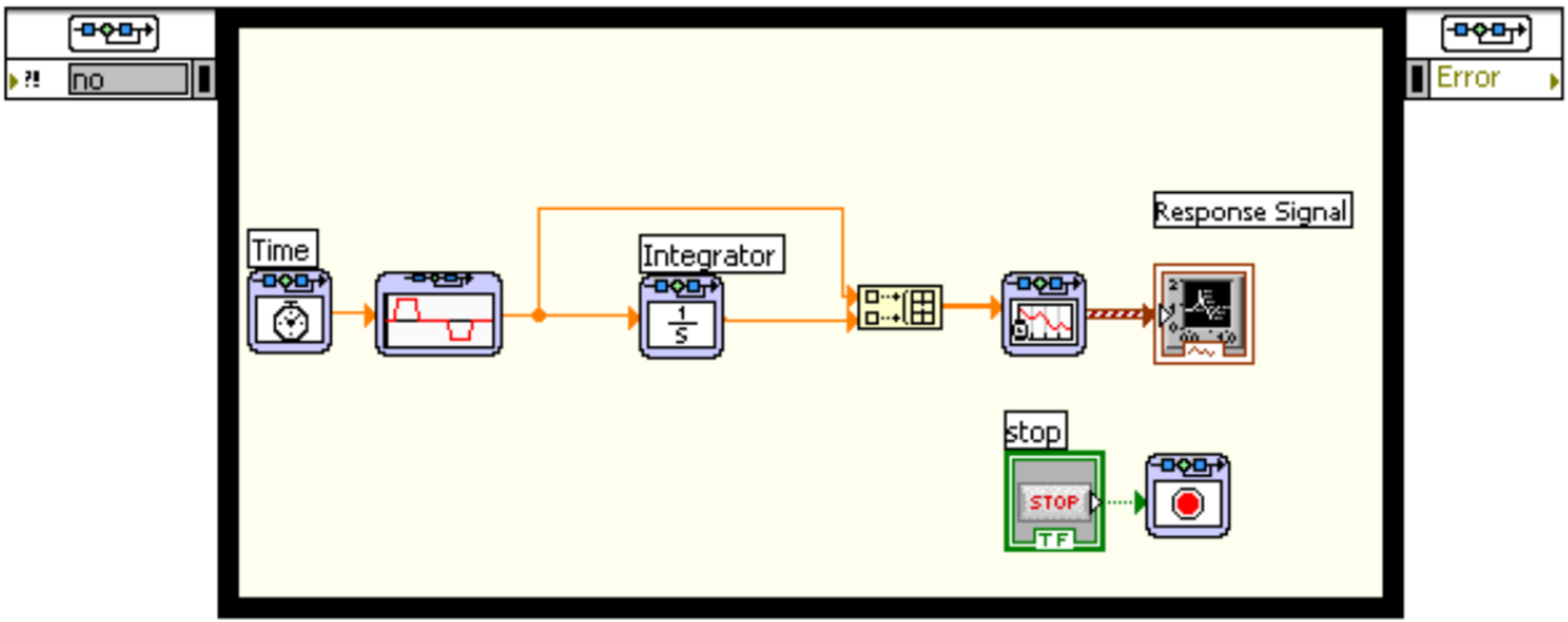
Процесна информация и обработка

Лекции №9+10

Контрол и симулация в LabVIEW

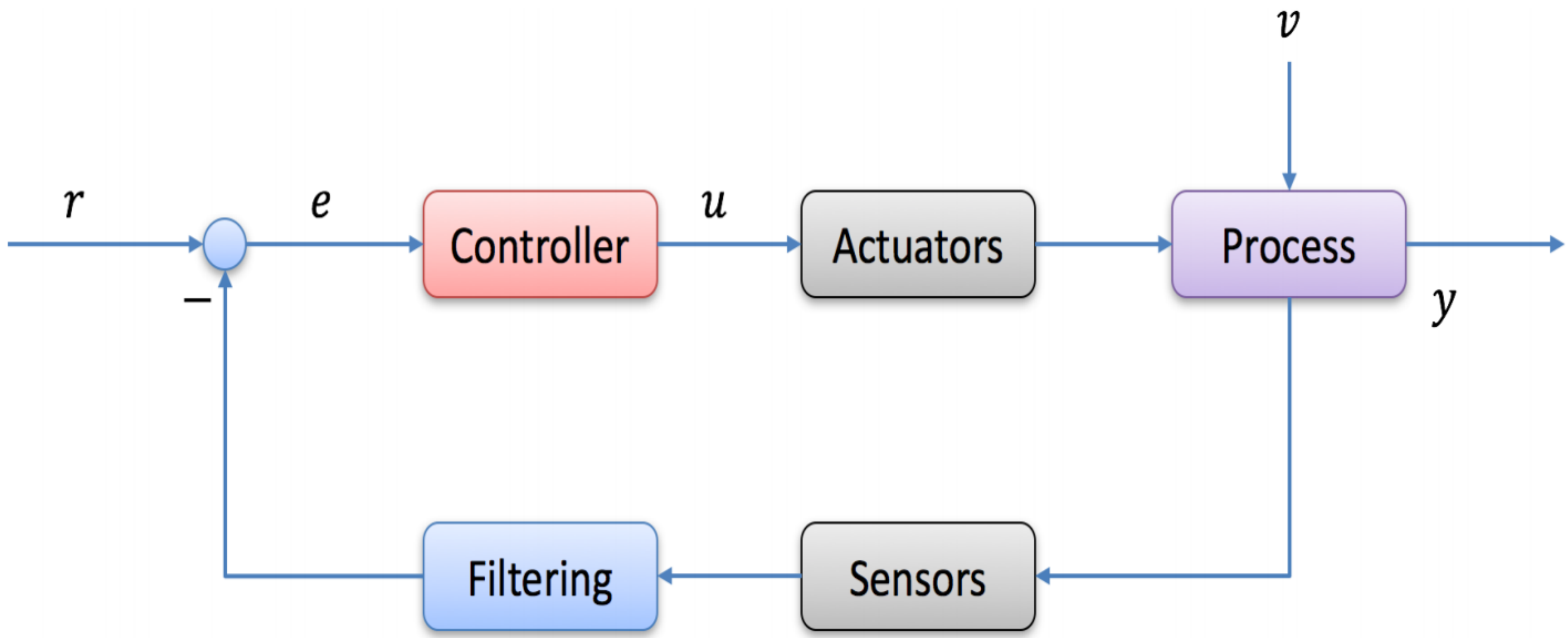


Основни понятия при използване на LabVIEW за контрол и симулация на технологични процеси.

Въведение

Контролът на управлението е процес, който включва разработването на математически модели, които описват целия физически процес и физическа система, като анализира моделите, за да се определят техните динамични характеристики и създаде контролер за постигане на определени динамични характеристики.

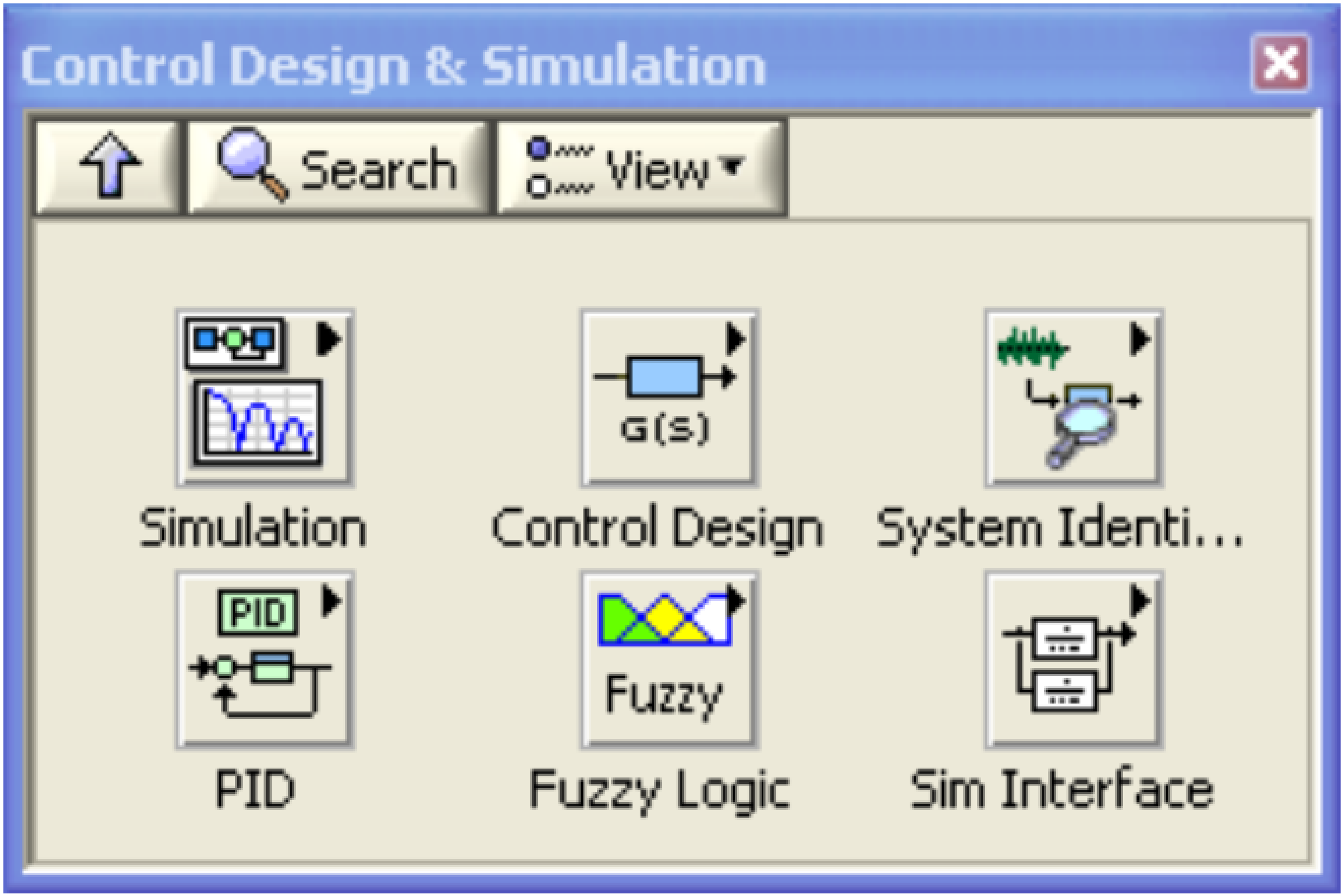
Симулацията е процес, който включва използването на софтуер за пресъздаване и анализиране на поведението на динамични системи. Симулационния процес се използва, за да се намалят разходите за разработка на продуктите и за ускоряване на разработването на продукти. Можете също така да се определи поведението на динамични системи, които не може да се възпроизведт удобно в лабораторията.



LabVIEW има няколко допълнителни модула и инструментариума за управление и симулация,

“LabVIEW Control Design and Simulation Module”,
“LabVIEW PID and Fuzzy LogicToolkit”,
“LabVIEW System Identification Toolkit”
“LabVIEW Simulation Interface Toolkit”.
LabVIEW MathScript.

Всички VI-и, свързани с тези модули и инструментариуми, са поставени в Control Design и Simulation Toolkit:



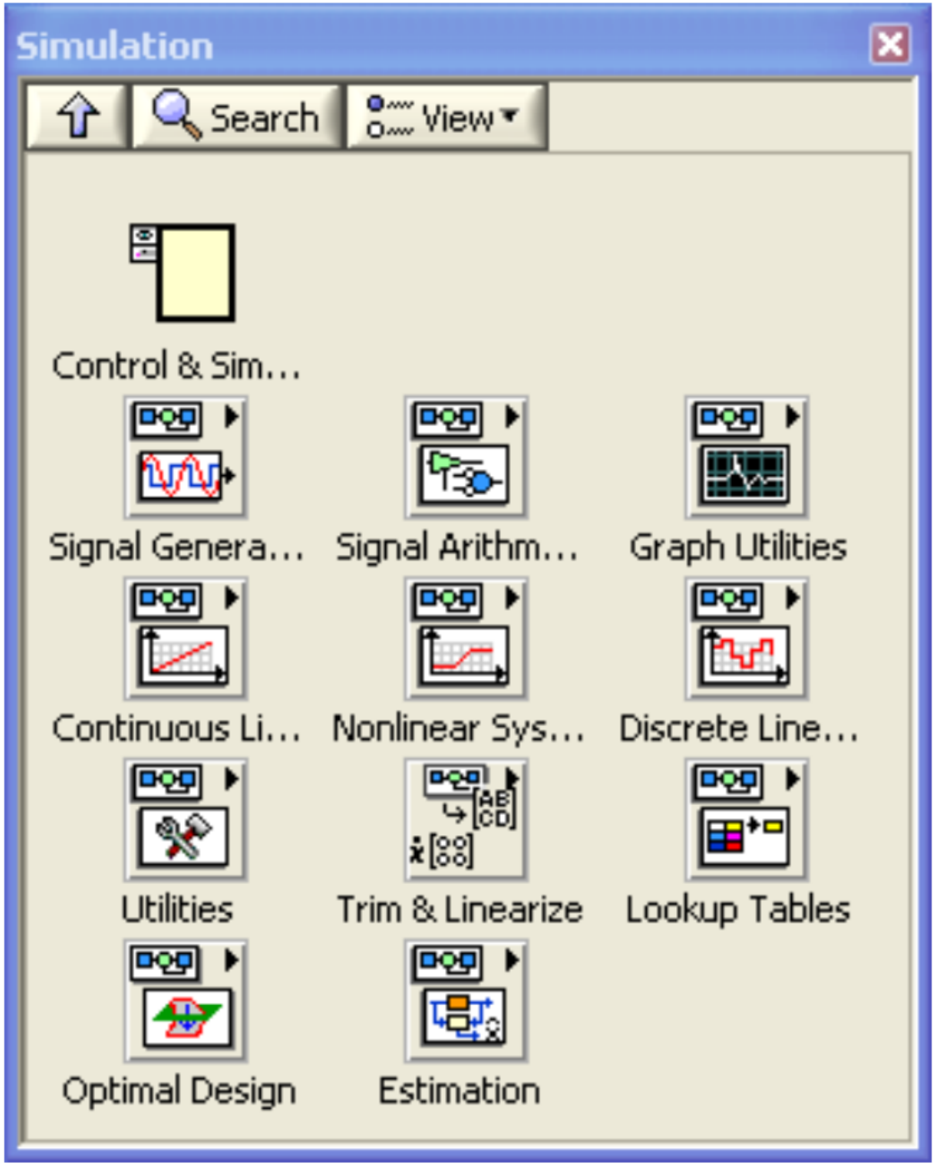
LabVIEW Control Design and Simulation Module

С LabVIEW контролния дизайн и симулационння модул можете да се изгради модел инсталация и контрол, използващи ***transfer function, state-space, or zero-pole-gain***.

Анализиране на ефективността на системата с ***step response, pole-zero maps, and Bode plots***

Симулиране на линейни, нелинейни, и дискретни системи с широк избор от решаващи методи. С дизайн NI LabVIEW Control Design и Симулационен модул, можете да анализирате поведението на модела с отворен цикъл, да проектирате затворен цикъл, контролери, симулират онлайн и офлайн системи и извършват физически реализации.

Simulation


Simulation
Search
View
Control & Sim...
Signal Genera...
Signal Arithm...
Graph Utilities
Continuous Li...
Nonlinear Sys...
Discrete Line...
Utilities
Trim & Linearize
Lookup Tables
Optimal Design
Estimation

Основните характеристики на палитрата на симулацията са:

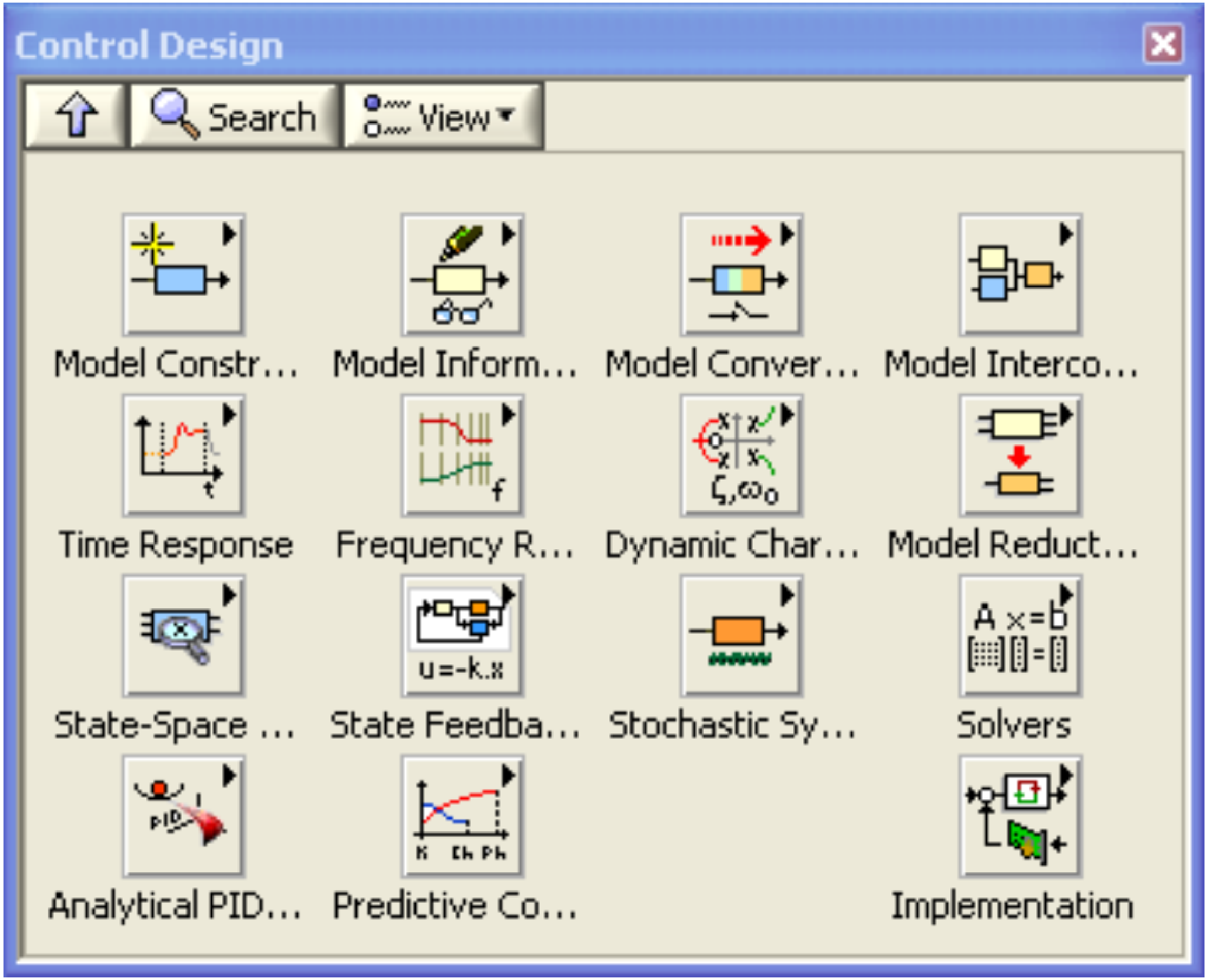
- Контрол и Симулация - ***Control and Simulation Loop***
- Функции на непрекъснати линейни системи – ***Continuous LinearSystems Functions***

Използване функциите за непрекъснати линейни системи за да се представят непрекъснати линейни системи на диференциални уравнения на симулацията диаграма.

- Аритметични функции на сигнала – ***Signal Arithmetic Functions***

Използване функциите на аритметиката на сигнала, за да изпълнение на основните аритметични операции на сигнали в симулационна система.

Control Design

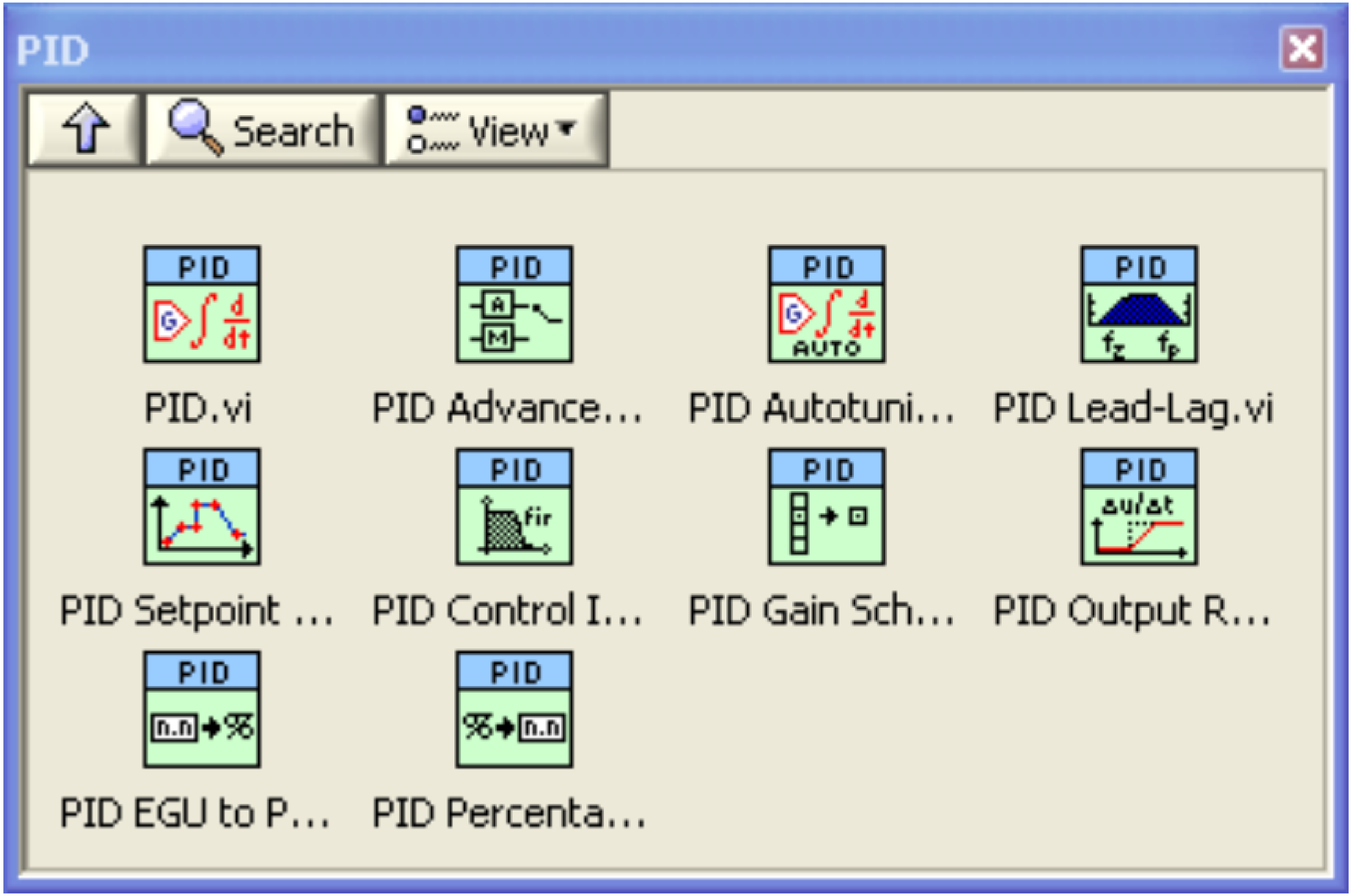


LabVIEW PID и Fuzzy Logic Toolkit

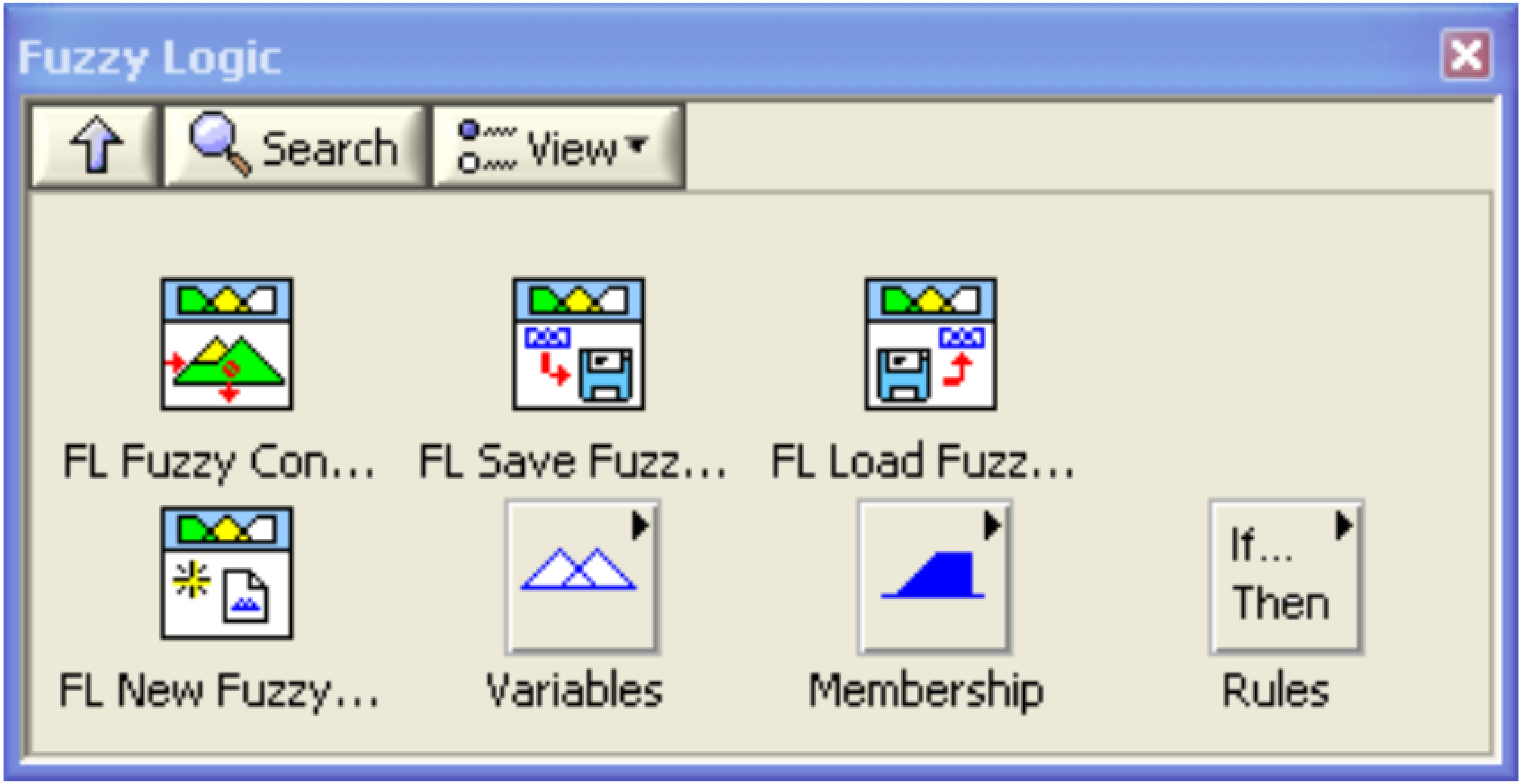
NI LabVIEW PID и Fuzzy Logic Toolkit добавят контролни алгоритми към LabVIEW. Чрез комбиниране на функциите за PID и размитите логически контроли в този инструментариум с математиката и логиката и функциите в софтуера LabVIEW, може бързо да се разработят програми за автоматично управление.

Можете да интегрирате тези контролни инструменти със система на събиране на данни.

PID Control


PID
Search
View
PID.vi
PID Advance…
PID Autotuni…
PID Lead-Lag.vi
PID Setpoint …
PID Control I…
PID Gain Sch…
PID Output R…
PID EGU to P…
PID Percenta…

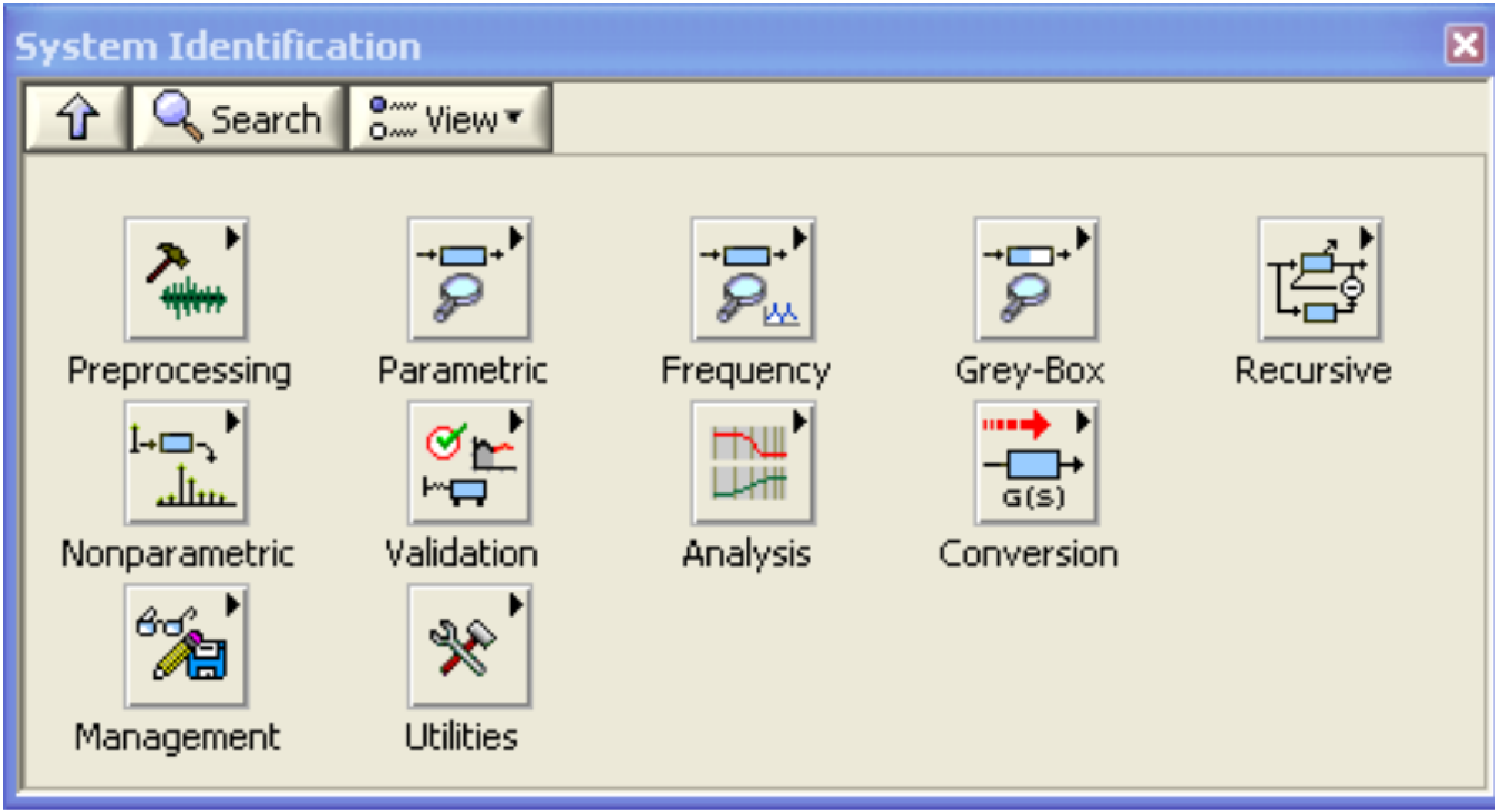
Fuzzy Logic



Fuzzy Logic
Search
View
FL Fuzzy Con...
FL Save Fuzz...
FL Load Fuzz...
FL New Fuzzy...
Variables
Membership
If...
Then
Rules

LabVIEW System Identification Toolkit

Инструмент за идентификация на системите LabVIEW
"LabVIEW System Identification Toolkit" съчетава инструментите за събиране на данни със системата идентификационни алгоритми за моделиране на процесите. Можете да използвате идентификацията на системата LabVIEW Наръчник за намиране на емпирични модели от реална информация за стимули-реакция на процесите.
Палети за идентификация на системата в LabVIEW:


System Identification
Search
View
Preprocessing
Parametric
Frequency
Grey-Box
Recursive
Nonparametric
Validation
Analysis
Conversion
G(s)
Management
Utilities

Симулация

Симулацията е процес, който включва използването на софтуер за пресъздаване и анализиране на поведението на

Динамични системи. Използвате симулационния процес, за да намалите разходите за разработка на продуктите

Ускоряване на разработването на продукти. Можете също така да използвате процеса на симулация, за да дадете прозрение

В поведението на динамични системи не можете да се възпроизвеждате удобно в лабораторията.

Например, симулиране на реактивен двигател спестява време, труд и пари в сравнение със строителството, Тестването и възстановяването на действителен реактивен двигател.

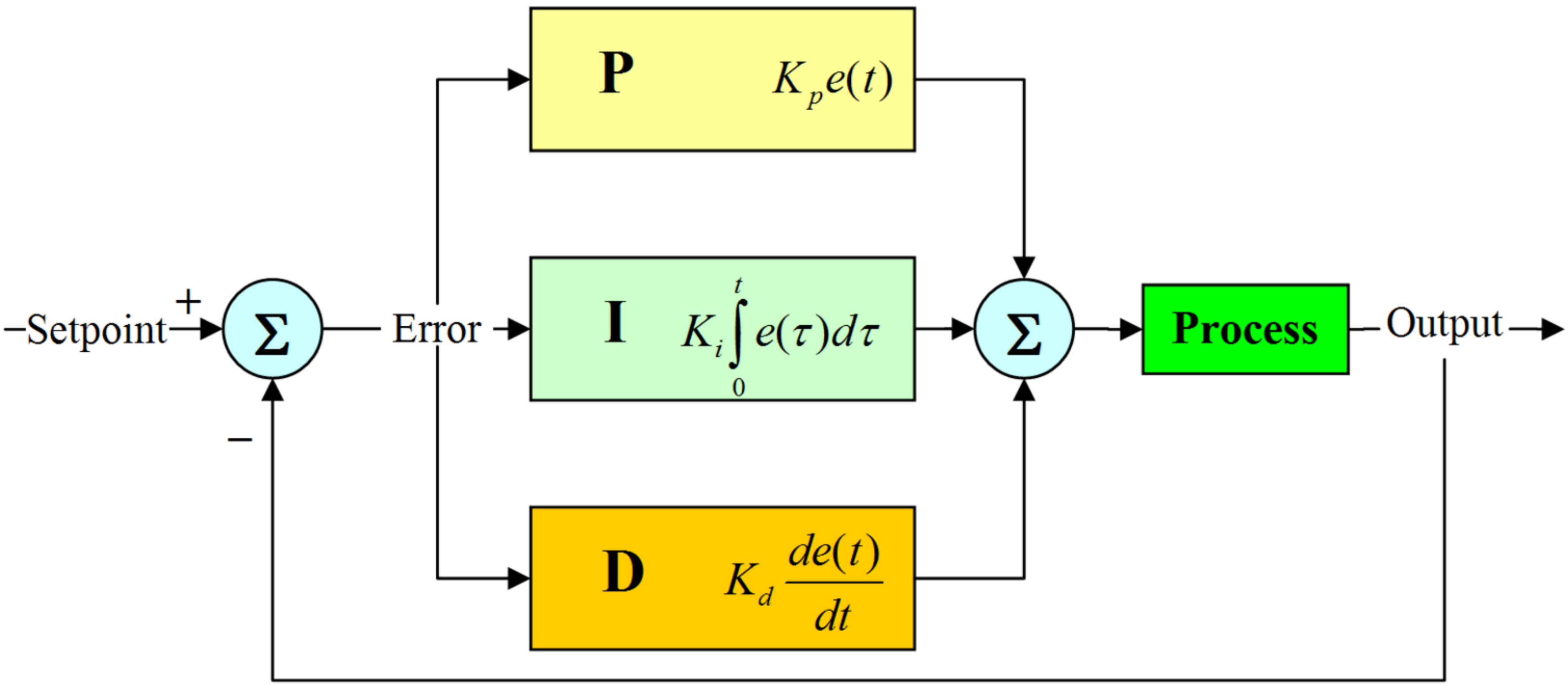
Можете да използвате LabVIEW Control Design и Симулационен модул за симулиране на динамична система или компонент на динамична система.

Например, можете да симулирате само централата, докато използвате хардуер за контролера, задвижванията, И сензори (симулация на хардуер в цикъл).

Динамичният модел на системата е диференциално или различно уравнение, което описва поведението На динамичната система.

PID Control

PID контролер се нарича всяко устройство или механизъм за управление, използващо обратна връзка и PID алгоритъм. Името му идва от първите букви на трите английски думи proportional, integral, derivative (пропорционален – интегрален – диференциален).



Използва се за управление на различни индустриални процеси като управление оборотите на двигатели, температурата на нагреваеми модели и др.

PID контролерът се стреми да поправи грешката, получена между предварително установената стойност и текущата стойност на контролираната величина (температура, ниво, обороти или нещо друго).
Грешката се получава като разлика между желаното състояние (напр. 60 градуса) и текущата измерена стойност. Измерванията могат да бъдат данни от тахогенератори, термистори и др.

$$u(t) = K_\mathrm{p} e(t) + K_\mathrm{i} \int_0^t e(\tau)\, d\tau + K_\mathrm{d} \frac{de(t)}{dt},$$

ефект от повишаване на коефициента:

коефициент	ускорение на нарастването	прехвърляне на зададената стойност	бързина на затихване	постоянна грешка (при линеен изход)
$K_{\text{проп.}}$	по-голямо	увеличава се	малка промяна	намалява
$K_{\text{инт.}}$	по-голямо	увеличава се	намалява	премахва се
$K_{\text{диф.}}$	малко по-голямо	намалява	увеличава се	няма ефект

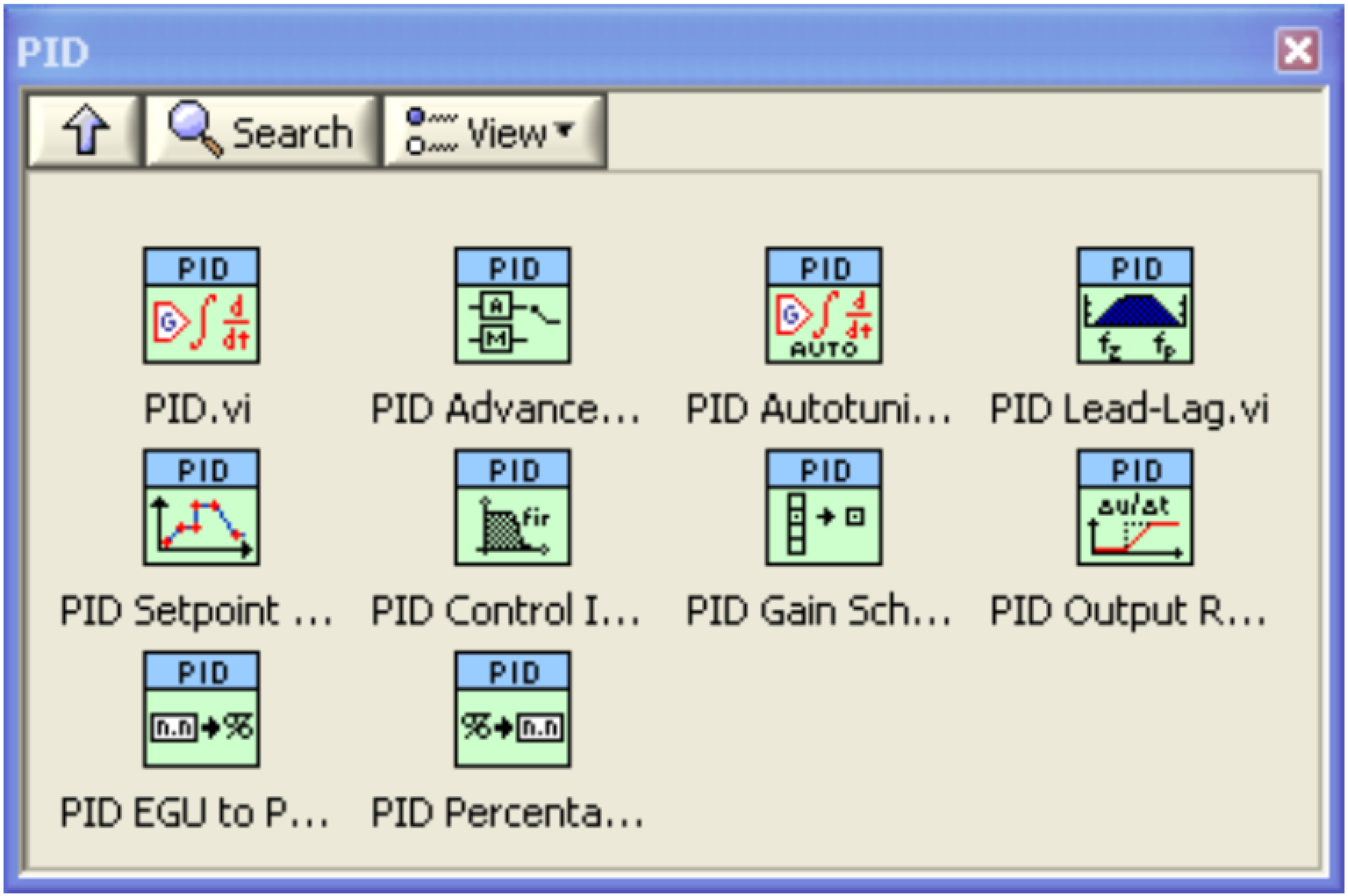
Корекцията се изчислява на базата на три части:
1. P – пропорционална – има смисъл на моменталната и директна реакция на контролера към грешката. Напр. ако двигателят се забавя – контролерът увеличава подаваното към него напрежение.
2. I – интегрална – смисълът ѝ е за по-бърза реакция към по-дълготрайни грешки. Напр. при начално установяване на зададените обороти.
3. D – диференциална – има смисъл за реакция към бързите промени на грешката и по-бързото затихване на грешката. Напр. когато температурата започне да се покачва, още преди да е стигнала желаното ниво контролерът да спре нагревателя, за да не се стигне до прегряване.

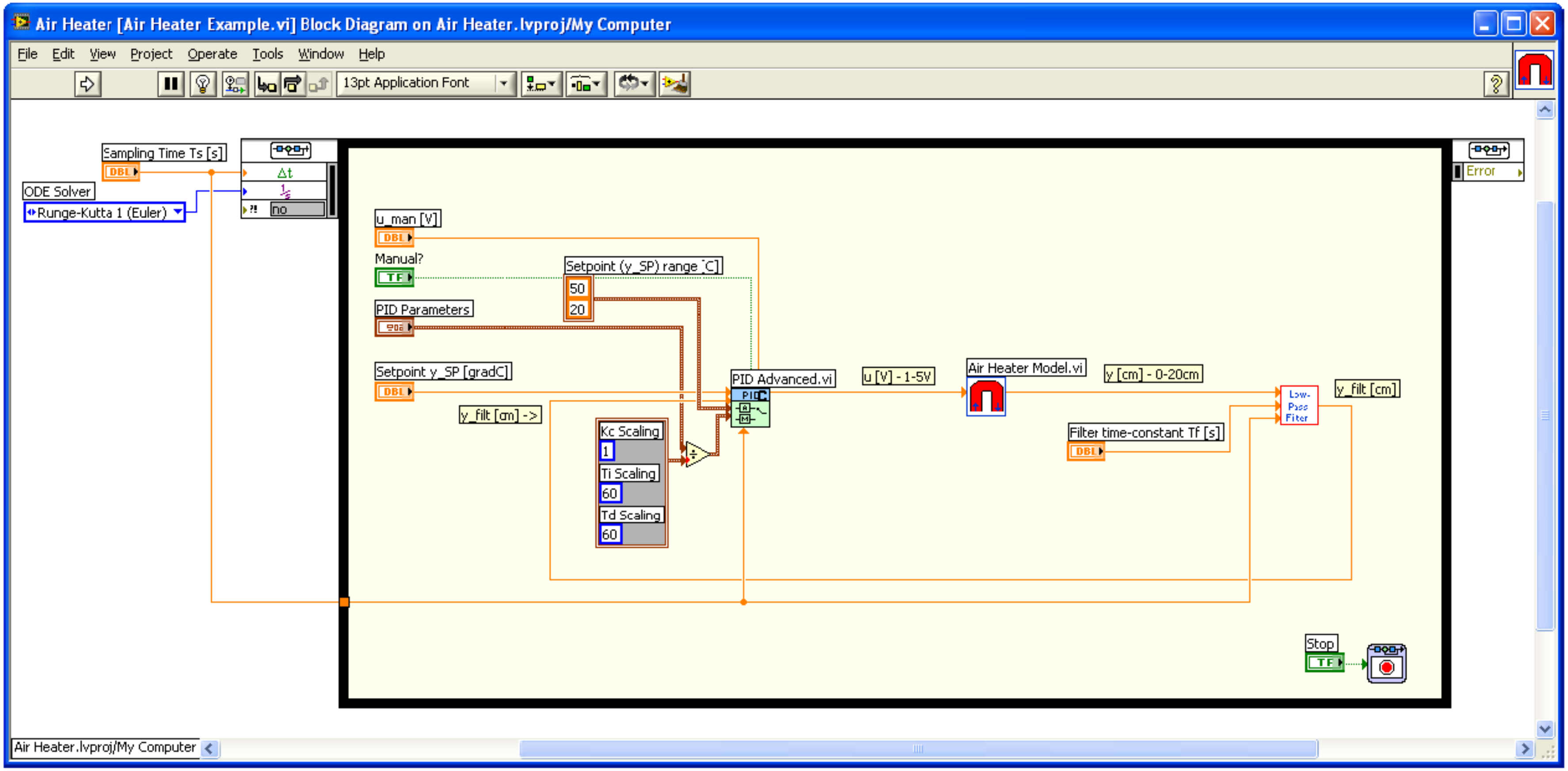
Всяка част се умножава с подходящ коефициент и частите се събират в една стойност, която определя реакцията на контролера.

Реакцията може да се предостави линейно към изхода (напрежение, честота и др.) или да премине през размита логика (Fuzzy Logic), която да определи какво да се прави (напр. – голямо повишение, малко повишение, без промяна, малко намаление, голямо намаление).

Възможно е някои от коефициентите да са 0. Тогава се получават разновидности на PID контролера (P, PI, PD, D, I, ID).

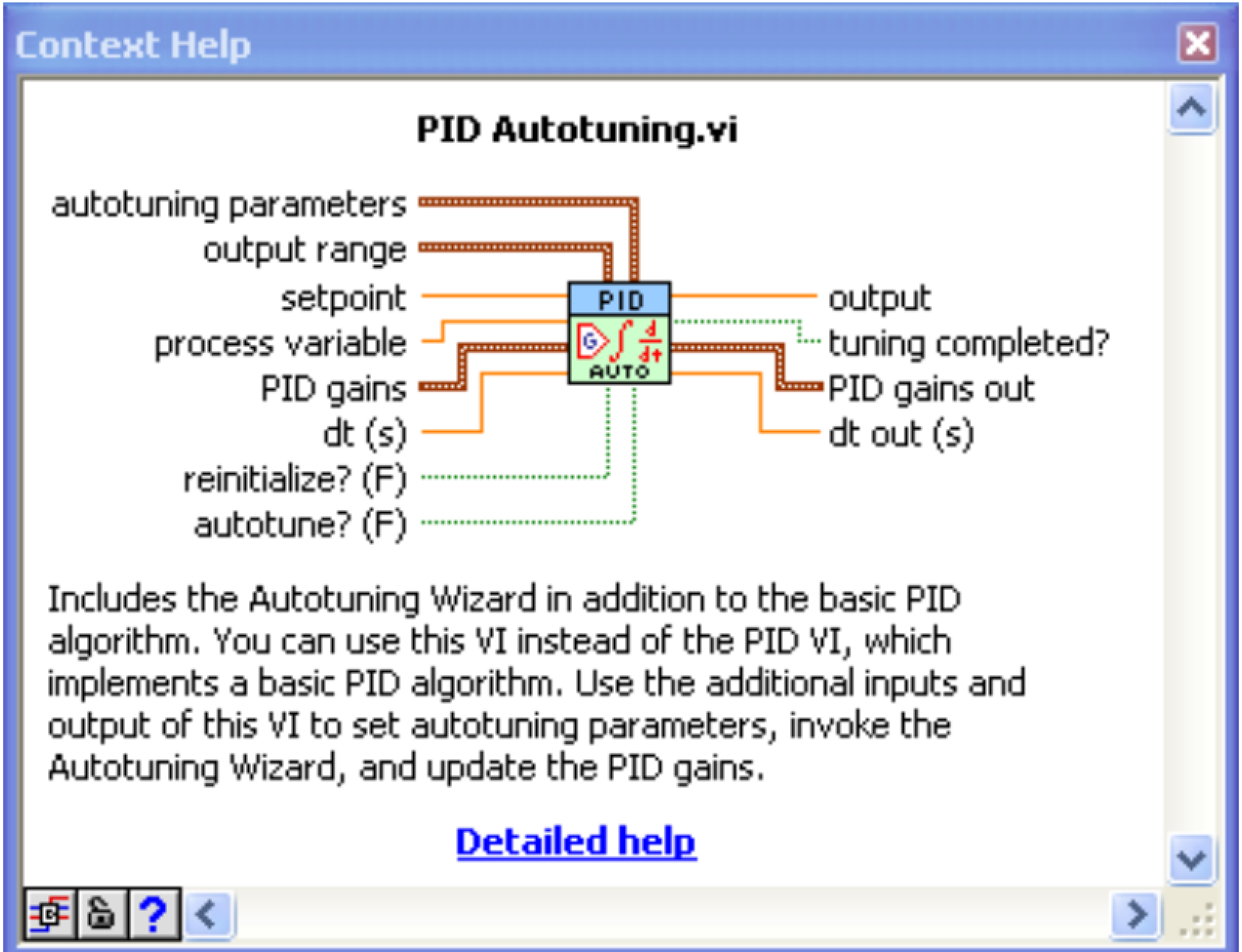
Коефициентите могат да се настройват по няколко начина: ръчно, метод на Ziegler–Nichols, със спец. софтуер за настройка, метод на Cohen-Coon.

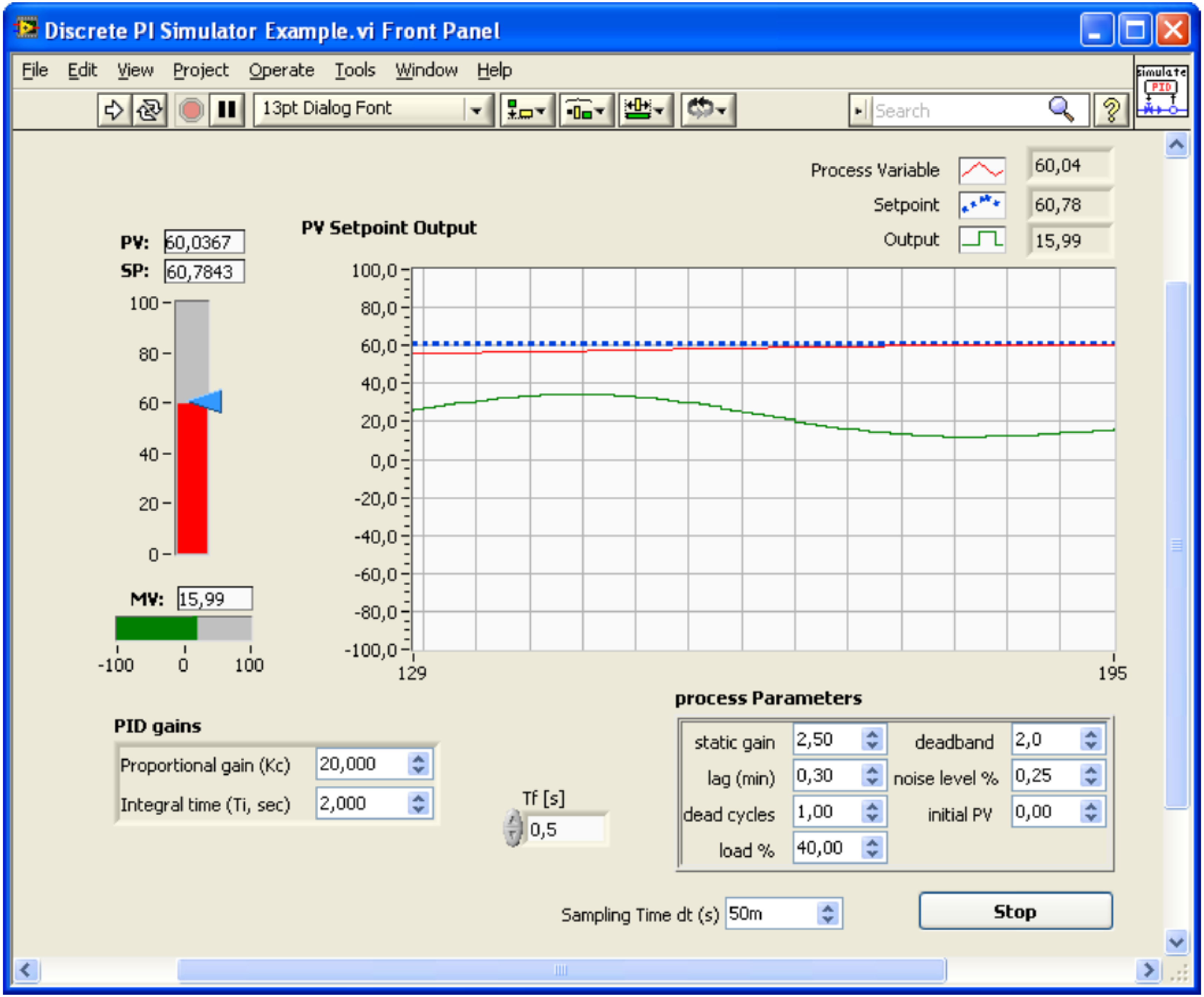

PID
Search
View
PID.vi
PID Advance...
PID Autotuni...
PID Lead-Lag.vi
PID Setpoint ...
PID Control I...
PID Gain Sch...
PID Output R...
PID EGU to P...
PID Percenta...



Air Heater [Air Heater Example.vi] Block Diagram on Air Heater.lvproj/My Computer
File Edit View Project Operate Tools Window Help
13pt Application Font
Sampling Time Ts [s]
ODE Solver
Runge-Kutta 1 (Euler)
no
u_man [V]
Manual?
Setpoint (y_SP) range [C]
50
20
PID Parameters
Setpoint y_SP [gradC]
y_filt [cm] ->
Kc Scaling
1
Ti Scaling
60
Td Scaling
60
PID Advanced.vi
u [V] - 1-5V
Air Heater Model.vi
y [cm] - 0-20cm
y_filt [cm]
Low-Pass Filter
Filter time-constant Tf [s]
Error
Stop
Air Heater.lvproj/My Computer

Auto-tuning


Context Help
PID Autotuning.vi
autotuning parameters
output range
setpoint
process variable
PID gains
dt (s)
reinitialize? (F)
autotune? (F)
PID
AUTO
output
tuning completed?
PID gains out
dt out (s)
Includes the Autotuning Wizard in addition to the basic PID algorithm. You can use this VI instead of the PID VI, which implements a basic PID algorithm. Use the additional inputs and output of this VI to set autotuning parameters, invoke the Autotuning Wizard, and update the PID gains.
Detailed help

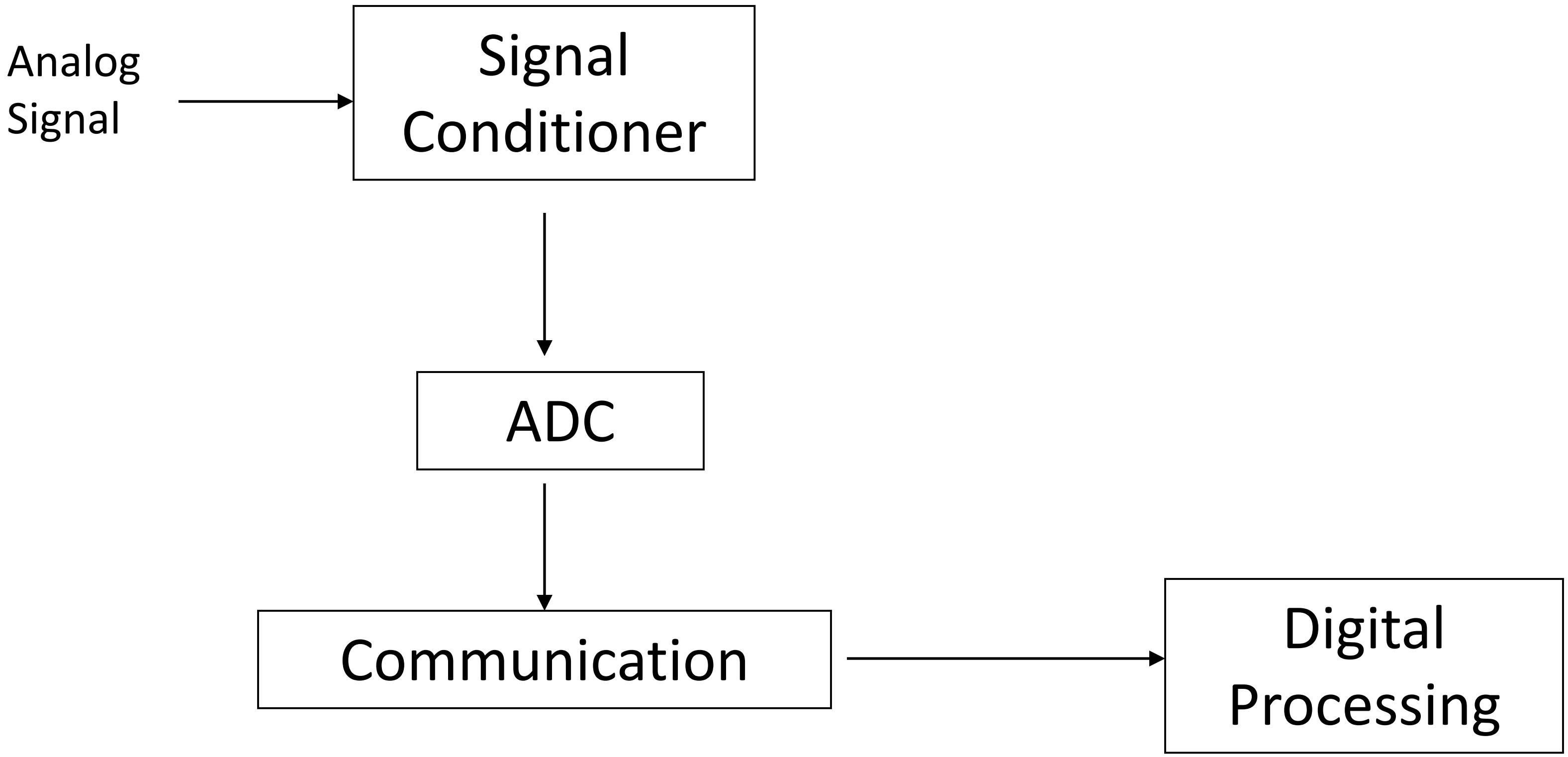


Discrete PI Simulator Example.vi Front Panel
File Edit View Project Operate Tools Window Help
13pt Dialog Font
Search
Process Variable 60,04
Setpoint 60,78
Output 15,99
PV: 60,0367
SP: 60,7843
PV Setpoint Output
MV: 15,99
PID gains
Proportional gain (Kc) 20,000
Integral time (Ti, sec) 2,000
Tf [s] 0,5
process Parameters
static gain 2,50
lag (min) 0,30
dead cycles 1,00
load % 40,00
deadband 2,0
noise level % 0,25
initial PV 0,00
Sampling Time dt (s) 50m
Stop

СИСТЕМИ ЗА СЪБИРАНЕ НА ДАННИ

Data Acquisition

СИСТЕМИ ЗА СЪБИРАНЕ
НА ДАННИ
Data Acquisition System



Преобразуване
Analog-to-Digital Conversion (ADC)

Конвертиране на аналогови в цифрови сигнали

- Sample & hold
- Quantization
- Coding

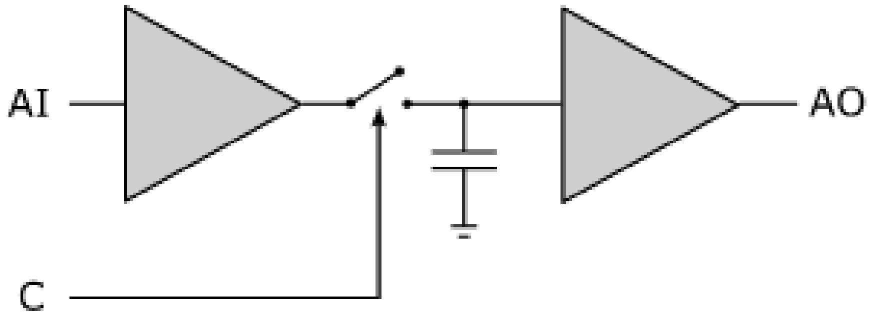
$$y(t)=f(t) \Rightarrow y_k=f(t_k)$$

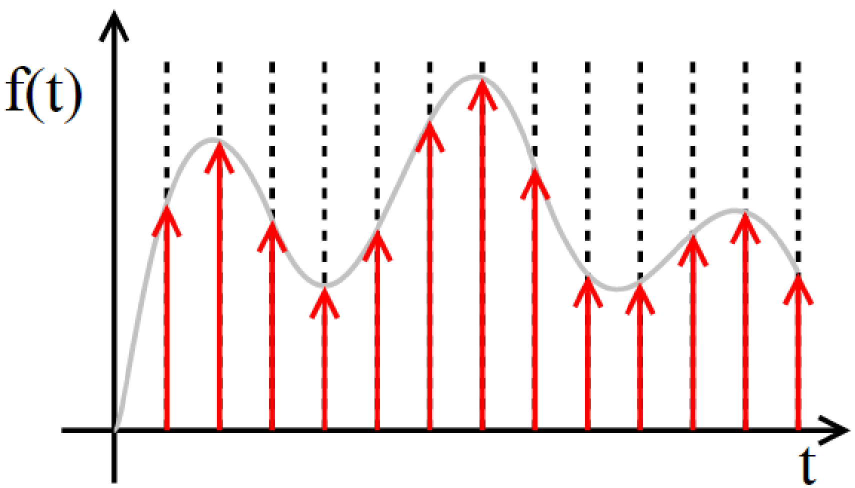
Sample and hold

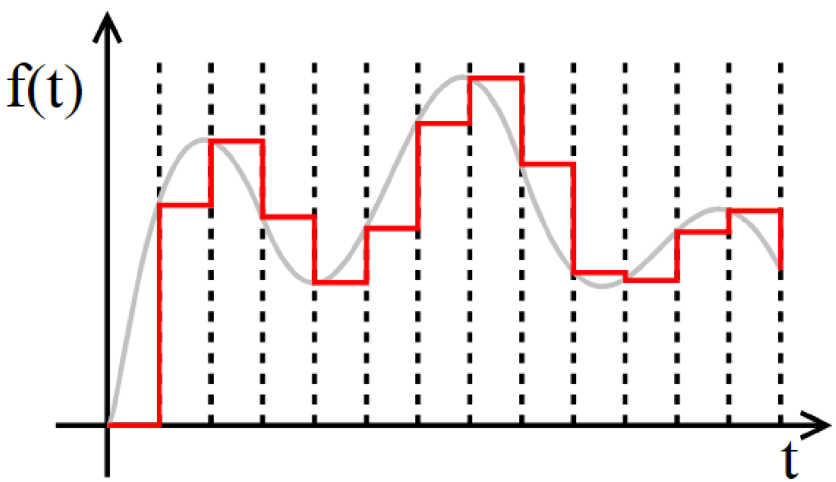
In electronics, a **sample and hold** (**S/H**, also "follow-and-hold”) circuit is an analog device that samples (captures, grabs) the voltage of a continuously varying analog signal and holds (locks, freezes) its value at a constant level for a specified minimum period of time. Sample and hold circuits and related peak detectors are the elementary analog memory devices. They are typically used in analog-to-digital converters to eliminate variations in input signal that can corrupt the conversion process.

A typical sample and hold circuit stores electric charge in a capacitor and contains at least one switching device such as a FET (field effect transistor) switch and normally one operational amplifier. To sample the input signal the switch connects the capacitor to the output of a buffer amplifier. The buffer amplifier charges or discharges the capacitor so that the voltage across the capacitor is practically equal, or proportional to, input voltage. In hold mode the switch disconnects the capacitor from the buffer. The capacitor is invariably discharged by its own leakage currents and useful load currents, which makes the circuit inherently volatile, but the loss of voltage (*voltage drop*) within a specified hold time remains within an acceptable error margin.

Пример за Sample and hold







Квантоване Quantization

Определение: трансформация на непрекъснат аналогов вход в набор от дискретни състояния на изхода

- Coding: the assignment of a digital code word or number to each output states
- # of possible state: $N=2^{n,}$ n is # of bits
- Quantization resolution: Q=(Vmax-Vmin)/N
- Quantization Error:

$$\Delta = \sum_{N} \left| f(t_k) - f_k \right|$$

Select a Data Acquisition Card

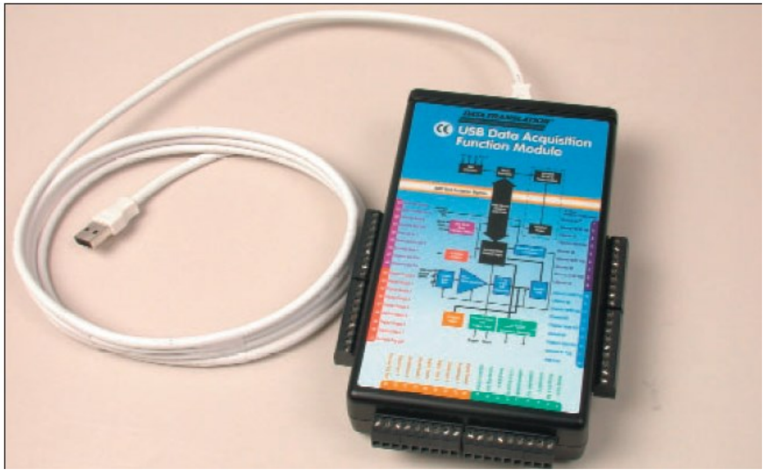
- Functions: A/D, D/A, Digital I/O, signal conditioning (amplification, prefiltering), timer, trigger, buffer
- Features:
  - A/D resolution (# of bits used)
  - Maximum sampling rate
  - # of channels
  - Total throughput
  - Aperture time

Example of Data Acquisition Card

Features Summary

Analog Inputs				
Board	**Channels**	**Resolution**	**Input Ranges**	**Throughput**
DT9801	16SE/8DI	12 bits	±1.25, 2.5, 5, 10 V 0-1.25, 2.5, 5, 10 V	100 kS/s
DT9802	16SE/8DI	12 bits	±1.25, 2.5, 5, 10 V 0-1.25, 2.5, 5, 10 V	100 kS/s
DT9803	16SE/8DI	16 bits	±1.25, 2.5, 5, 10 V	100 kS/s
DT9804	16SE/8DI	16 bits	±1.25, 2.5, 5, 10 V	100 kS/s

	Analog Outputs				Digital I/O	
Board	**Channels**	**Resolution**	**Output Ranges**	**Output Speed**	**I/O Lines**	**Counter/Timer**
DT9801	0	NA	NA		16	2
DT9802	2	12 bits	±5, 10, 0-5, 0-10 V	50Hz	16	2
DT9803	0	NA	NA		16	2
DT9804	2	16 bits	±10 V	50Hz	16	2



Timing

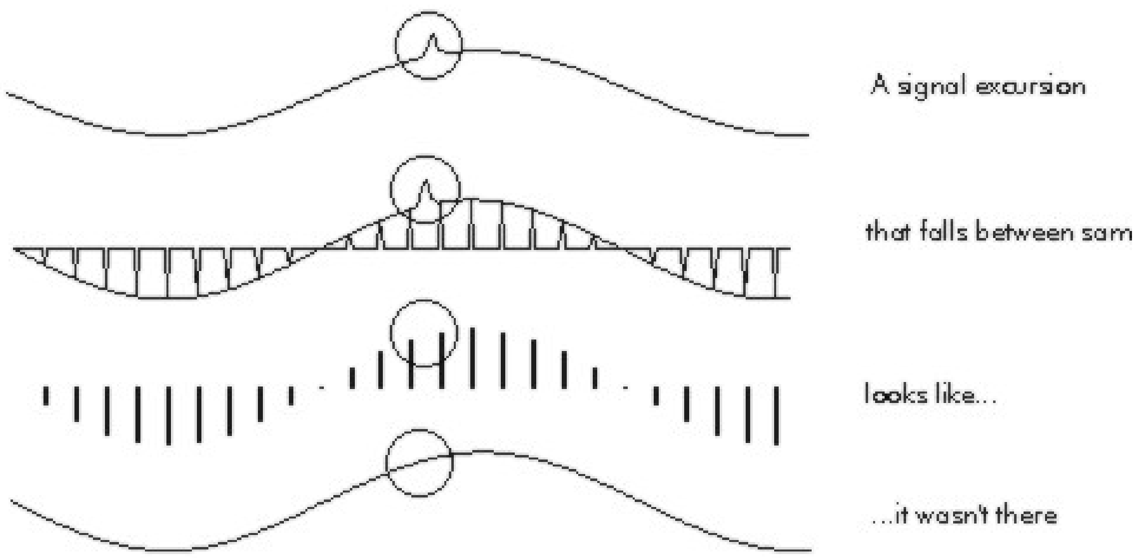
- Aperture time: the duration of the time window that the analog is sampled
- Jitter:

Sampling

- Sampling: Numerical evaluate the signal at discrete distance in time, $y_k=y(k\Delta t)$
- Digitized Signal: a sequence of numbers that is an approximation to an analog signal
- Sampling time/Period: time duration between two consecutive samples, Δt
- Sampling rate (Hz): $1/\Delta t$
- Nyquist Frequency: $2f_{max}$
- Sampling theory: fs > Nyquist Frequency

Sampling Theory

- Shannon-Nyquist sampling theorem
  - The maximum frequency component a sampled data system can accurately handle is its Nyquist limit (i.e., Nyquist frequency).



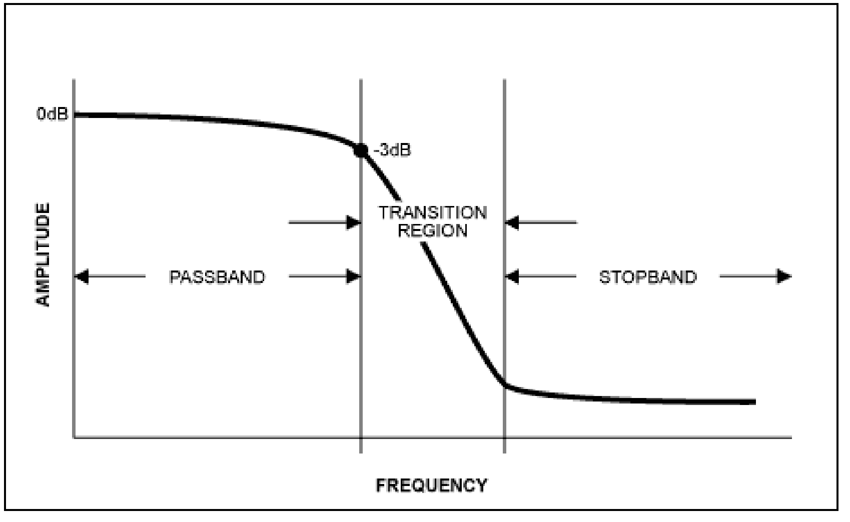
Aliasing

- Matlab example of aliasing

на погрешно идентифициране честотен сигнал, въвеждане на изкривяване или грешка.
Аналоговите сигнали обикновено са нискочестотен филтрува за отстраняване на повечето или всички компоненти над честотата Nyquist за да се избегне заглаждане.

Anti-aliasing Filter

- One way of avoiding the problem of aliasing is to apply an anti-aliasing filter to the signal, prior to the sampling stage, to remove any frequency components above the "folding" or Nyquist frequency (half the sampling frequency).
- An anti-aliasing filter is a low-pass filter.



PC-based Data Acquisition System Overview

- In the last few years, industrial PC I/O interface products have become increasingly reliable, ccurate and affordable. PC-based data acquisition and control systems are widely used in industrial and laboratory applications like monitoring, control, data acquisition and automated testing.
- Selecting and building a DA&C (Data Acquisition and Control) system that actually does what you want it to do requires some knowledge of electrical and computer engineering.
    - • Transducers and actuators
    - • Signal conditioning
    - • Data acquisition and control hardware
    - • Computer systems software

Data Acquisition System Introduction I

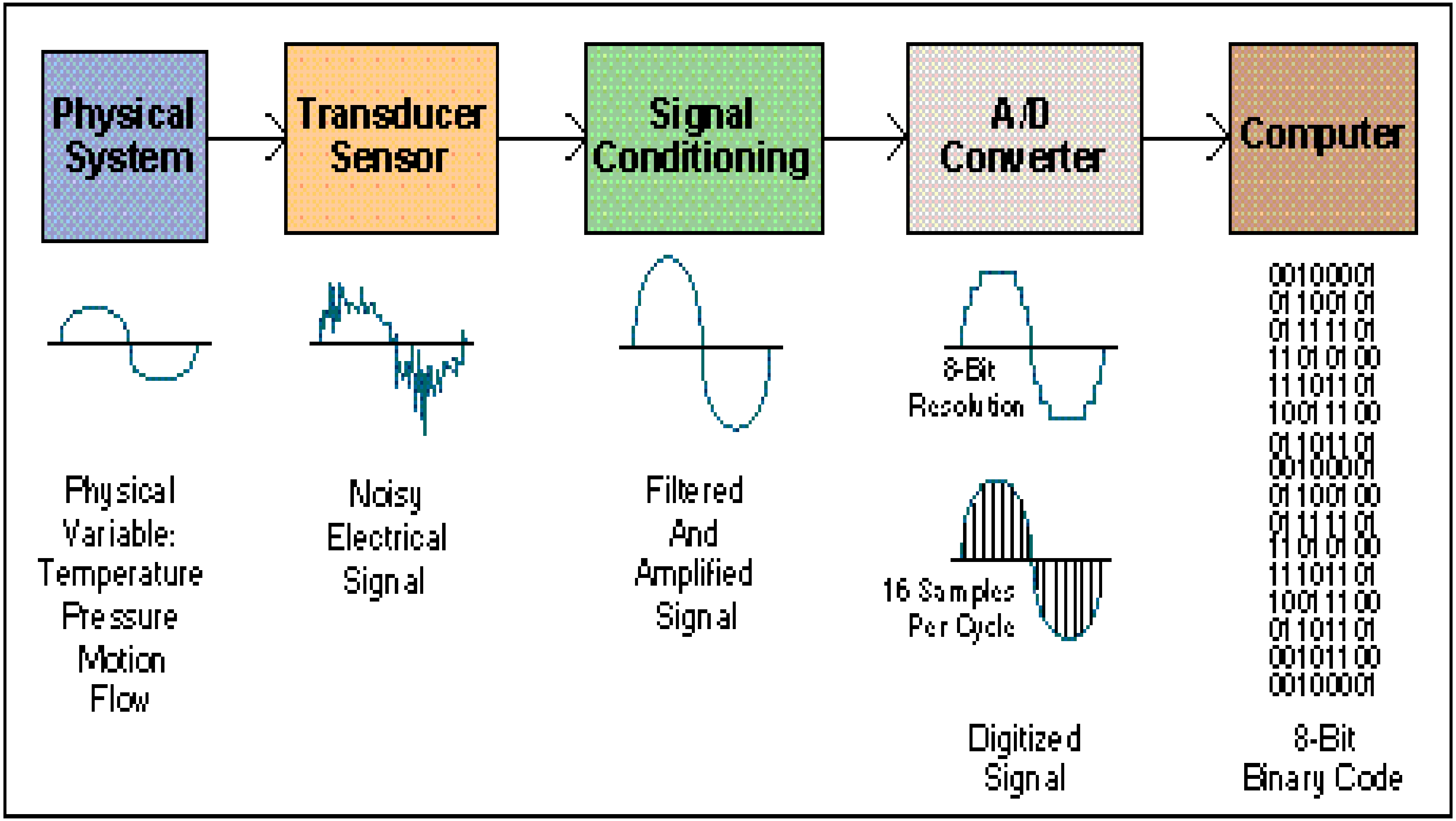
A data acquisition system consists of many components that are integrated to:

- Sense physical variables (use of transducers)
- Condition the electrical signal to make it readable by an A/D board

Data Acquisition System Introduction II

- Convert the signal into a digital format acceptable by a computer
- Process, analyze, store, and display the acquired data with the help of software

Data Acquisition System Block Diagram



Transducers

Sense physical phenomena and translate it into electric signal.

- Temperature
- Pressure
- Light
- Force
- Displacement
- Level
- Electric signals
- ON/OFF switch

Transducers and Actuators

- A transducer converts temperature, pressure, level, length, position, etc. into voltage, current, frequency, pulses or other signals.
- An actuator is a device that activates process control equipment by using pneumatic, hydraulic or electrical power. For example, a valve actuator opens and closes a valve to control fluid rate.

Signal Conditioning

- Signal conditioning circuits improve the quality of signals generated by transducers before they are converted into digital signals by the PC's data-acquisition hardware.
- Examples of signal conditioning are signal scaling, amplification, linearization, cold-junction compensation, filtering, attenuation, excitation, common-mode rejection, and so on.

Signal Conditioning

- One of the most common signal conditioning functions is amplification.
- For maximum resolution, the voltage range of the input signals should be approximately equal to the maximum input range of the A/D converter. Amplification expands the range of the transducer signals so that they match the input range of the A/D converter. For example, a x10 amplifier maps transducer signals which range from 0 to 1 V into the range 0 to 10 V before they go into the A/D converter.

Signal Conditioning

Electrical signals are conditioned so they can be used by an analog input board. The following features may be available:

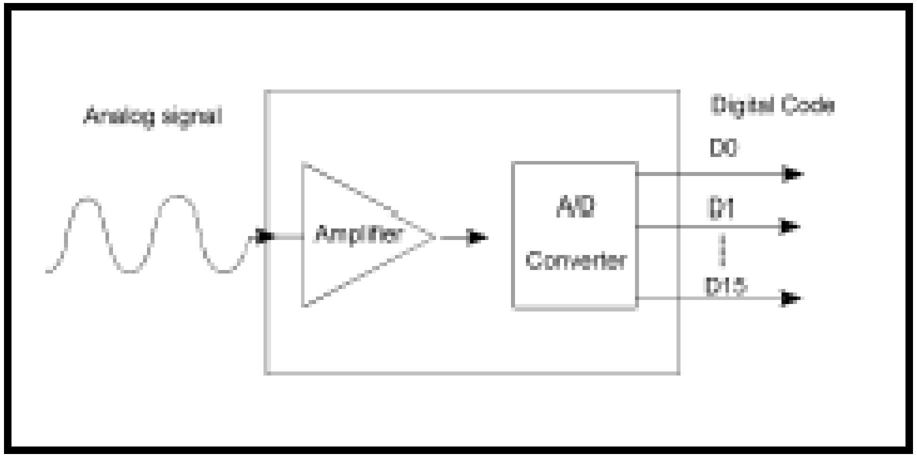
- Amplification
- Isolation
- Filtering
- Linearization

Data Acquisition

- Data acquisition and control hardware generally performs one or more of the following functions:
  - analog input,
  - analog output,
  - digital input,
  - digital output and
  - counter/timer functions.

Analog Inputs (A/D)

- Analog to digital (A/D) conversion changes analog voltage or current levels into digital information. The conversion is necessary to enable the computer to process or store the signals.



Analog Inputs (A/D)

- The most significant criteria when selecting A/D hardware are:
  - 1. Number of input channels
  - 2. Single-ended or differential input signals
  - 3. Sampling rate (in samples per second)
  - 4. Resolution (usually measured in bits of resolution)
  - 5. Input range (specified in full-scale volts)
  - 6. Noise and nonlinearity

Analog to Digital (A/D) Converter

- Input signal
- Sampling rate
- Throughput

- Resolution
- Range
- Gain

A/D Converter:
Input Signal

- Analog
  - ✓Signal is continuous

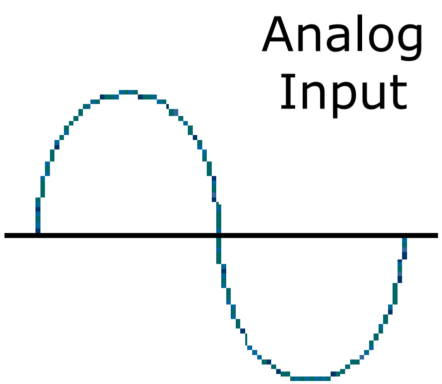
    Example: strain gage. Most of transducers produce analog signals

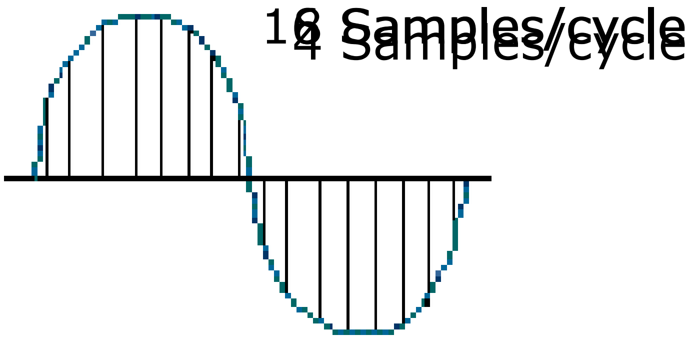
- Digital
  - ✓Signal is either ON or OFF

    Example: light switch.

A/D Converter:
Sampling Rate

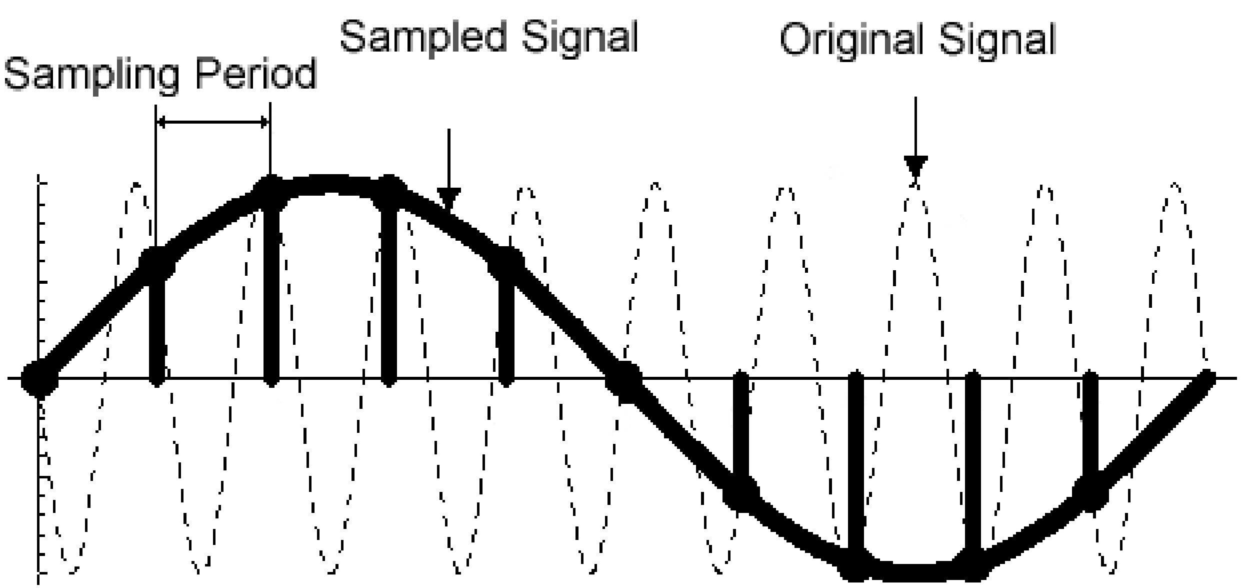
- Determines how often conversions take place.
- The higher the sampling rate, the better.





A/D Converter:
Sampling Rate

- Aliasing.
  - ✓ Acquired signal gets distorted if sampling rate is too small.



A/D Converter:
Throughput

Effective rate of each individual channel is inversely proportional to the number of channels sampled.

Example:

- 100 KHz maximum.
- 16 channels.

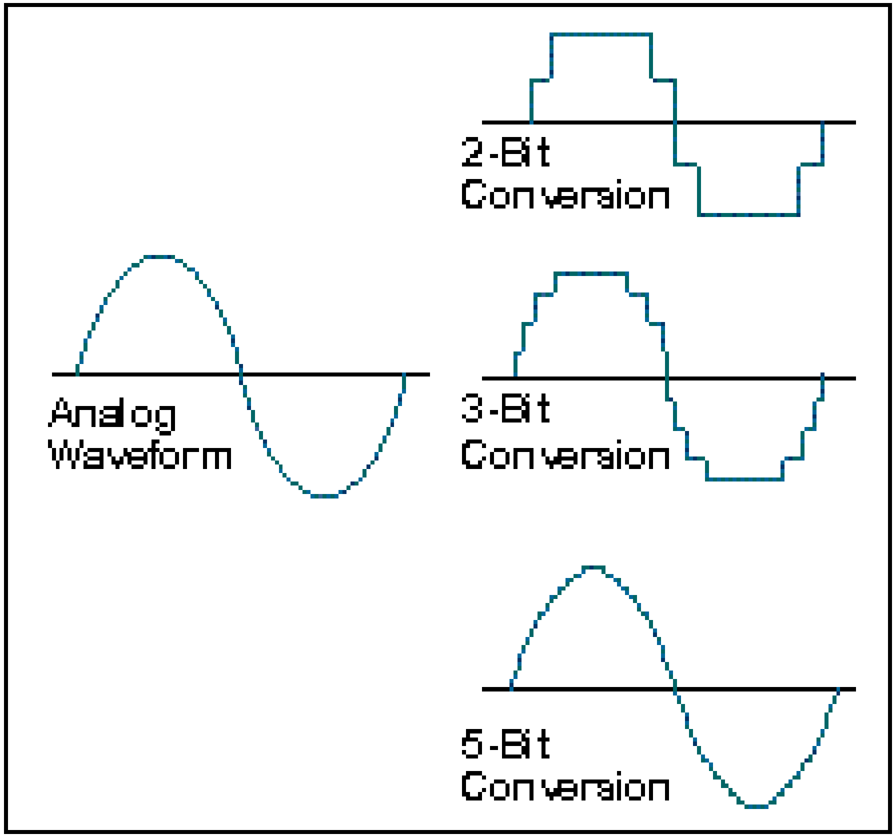
100 KHz/16 = 6.25 KHz per channel.

A/D Converter:
Range

- Minimum and maximum voltage levels that the A/D converter can quantize

- Ranges are selectable (either hardware or software) to accurately measure the signal

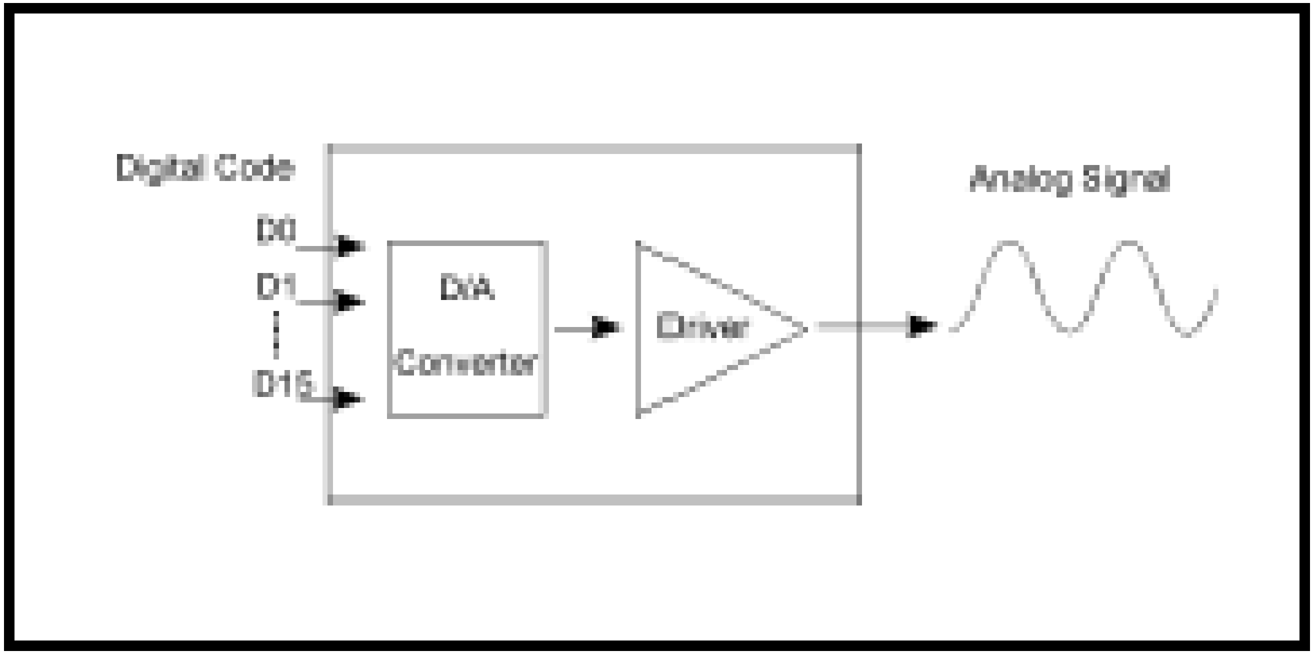
A/D Converter:
Resolution



Analog Outputs (D/A)

- The opposite of analog to digital conversion is digital to analog (D/A) conversion. This operation converts digital information into analog voltage or current. D/A devices allow the computer to control real-world events.
- Analog output signals may directly control process equipment. The process can give feedback in the form of analog input signals. This is referred to as a closed loop control system with PID control.
- Analog outputs can also be used to generate waveforms. In this case, the device behaves as a function generator.

Analog Outputs (D/A)



Data Acquisition Software

- It can be the most critical factor in obtaining reliable, high performance operation.
- Transforms the PC and DAQ hardware into a complete DAQ, analysis, and display system.
- Different alternatives:
  - Programmable software.
  - Data acquisition software packages.

Programmable Software

- Involves the use of a programming language, such as:
  - C++, visual C++
  - BASIC, Visual Basic + Add-on tools (such as VisuaLab with VTX)
  - Fortran
  - Pascal

✓ <u>Advantage</u>: flexibility

✓ <u>Disadvantages</u>: complexity and steep learning curve

Data Acquisition Software

- Does not require programming.
- Enables developers to design the custom instrument best suited to their application.

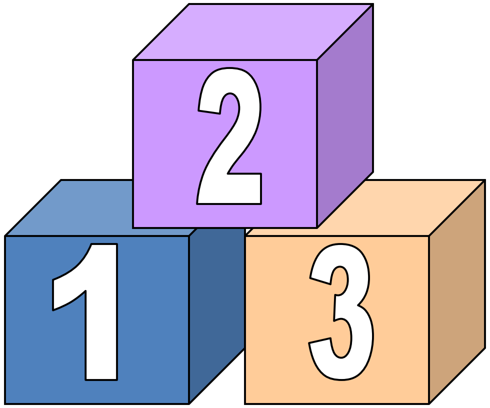
  Examples: TestPoint, SnapMaster, LabView, DADISP, DASYLAB, etc.

Designing a DAS: Factors to Consider

- Is it a fixed or a mobile application?
- Type of input/output signal: digital or analog?
- Frequency of input signal ?
- Resolution, range, and gain?
- Continuous operation?
- Compatibility between hardware and software. Are the drivers available?
- Overall price.

National Instruments LabVIEW and Data Acquisition: Applications for FIRST

Agenda



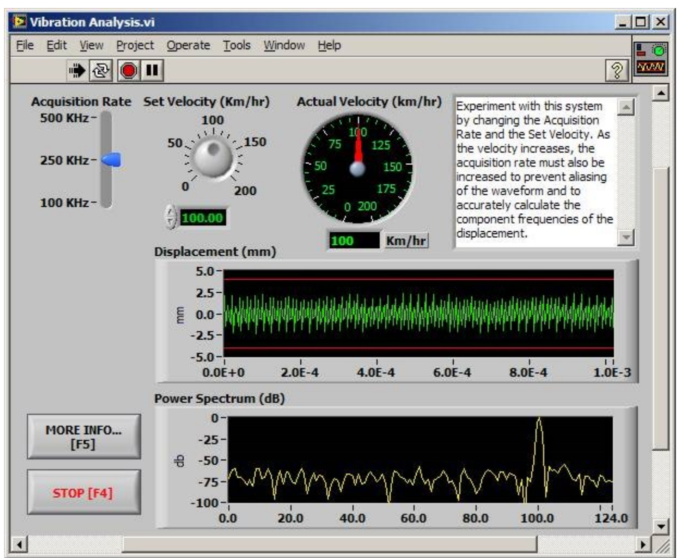
- Learn about LabVIEW
  - What is LabVIEW?
  - Why LabVIEW?
- Learn about NI-DAQ
  - What is Data Acquisition?
  - Introducing the NI USB-6009
  - How can USB-DAQ help me in *FIRST*?
- Demos

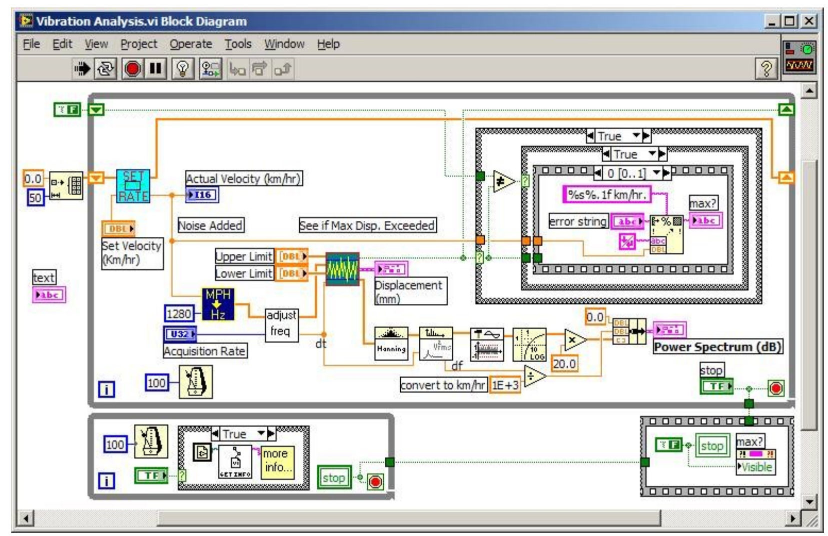
What is LabVIEW?

- LabVIEW - **Lab**oratory **V**irtual **I**nstrumentation **E**ngineering **W**orkbench
  - A graphical development environment (“G”)
  - A dataflow language
  - Has native multi-threaded parallel execution
  - A hardware communication/control platform
  - Can target real-time and embedded hardware

LabVIEW for Virtual Instrumentation

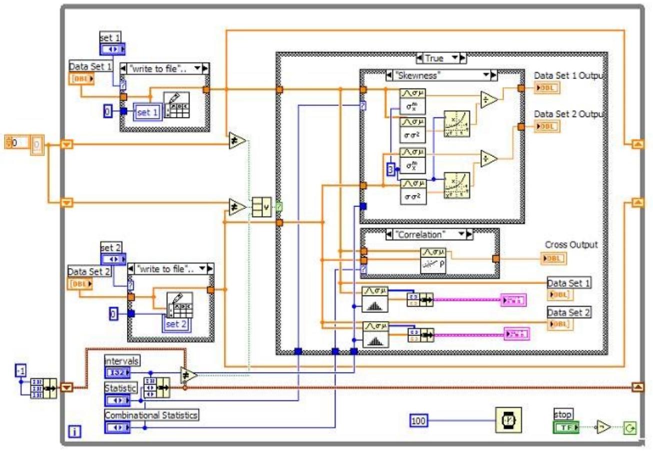
- **Virtual Instrument** – “Software that controls external hardware and displays data or results from that hardware to a front panel”





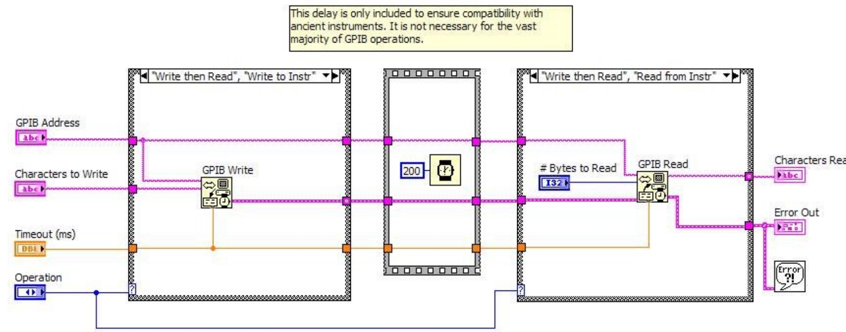
Graphical Programming

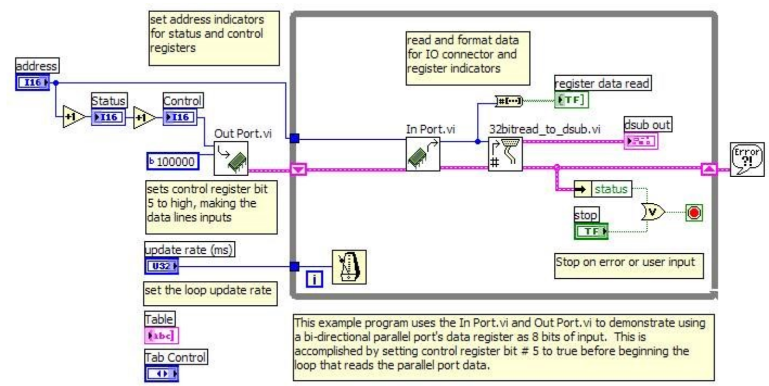
- LabVIEW – A native multi-threaded dataflow programming environment
  - Many of the same programming constructs as traditional programming language
  - Visual, easy to understand
  - MFC experience not a requirement



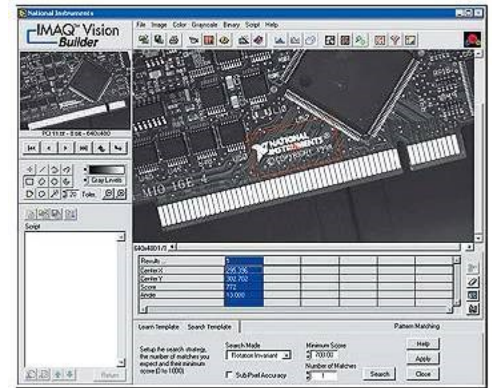
Hardware Communication

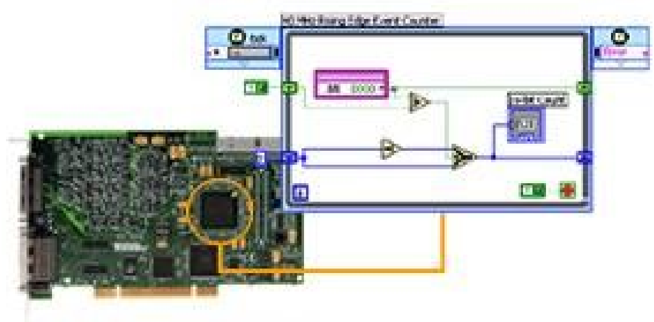
- Communicate over standard protocols
  - GPIB
  - Serial
  - Ethernet
- Control Devices via Register Mappings





Target Real-Time and Embedded Systems



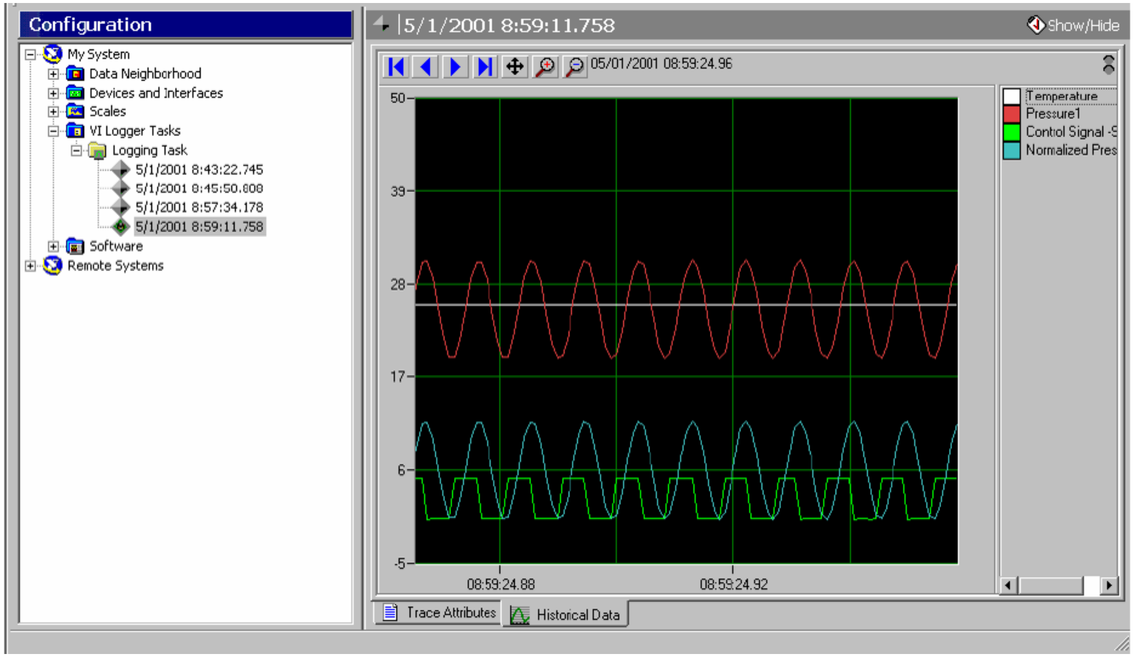


Why LabVIEW?

- LabVIEW as a multifunction environment
  - Rapid Prototyping
  - Out-of-the-box hardware integration
  - Near-seamless configuration
  - Portability of code
  - Graphical debugging

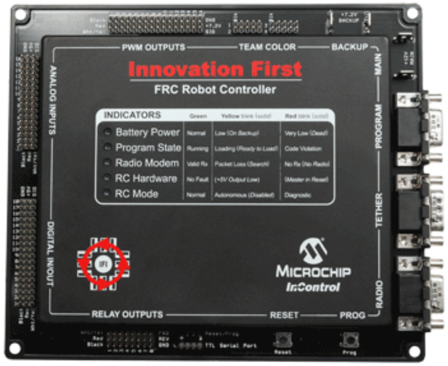
Introduction to DAQ

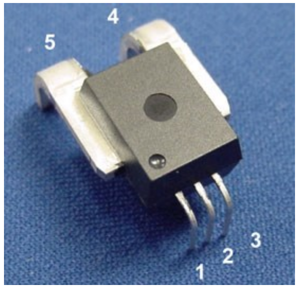
- Data Acquisition – “Sampling of the real world to generate data that can be analyzed and presented by a computer.”



DAQ in *FIRST Robotics*

- Integrated into FRC Controller
  - Analog Measurements
    - Battery voltage
    - Gyroscope data
    - Accelerometer data
    - Temperature measurements
    - Motor current measurements
    - Potentiometer positions
  - Digital Measurements
    - Hall-Effect sensors (gear-tooth sensors)
    - Reed switches
    - Limit Switches



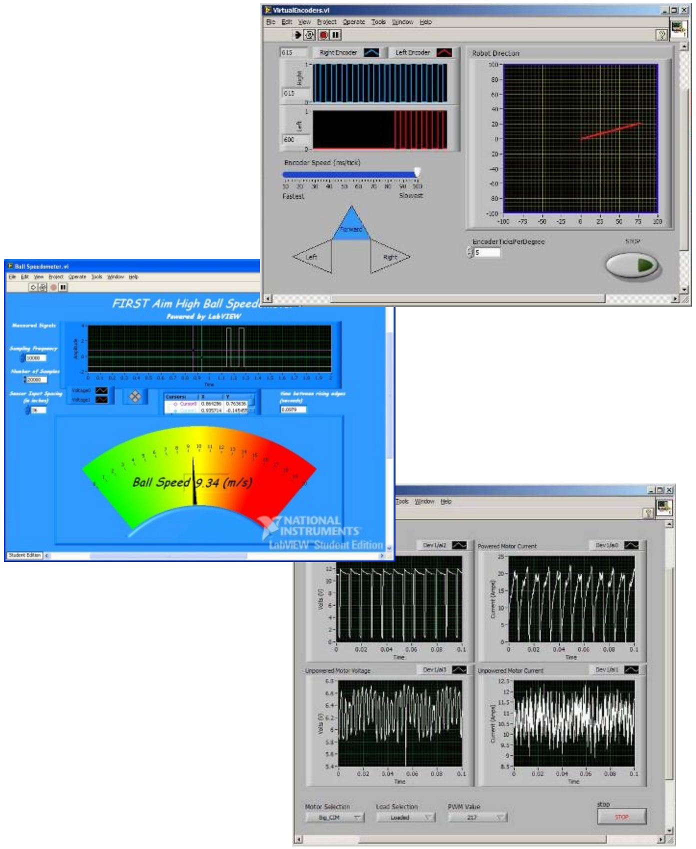


Introducing the NI USB-6009

- Low-Cost USB Data Acquisition
  - Windows/OSX/Linux Support
  - 8 Analog Input, 2 Analog Output
  - 12 Digital Lines
  - 1 Counter
  - 48 kS/s at 14-bit resolution
- Coming Soon to FRC Kits!

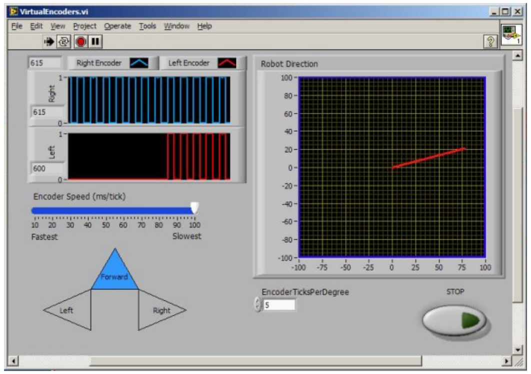
NI USB-6009 Uses

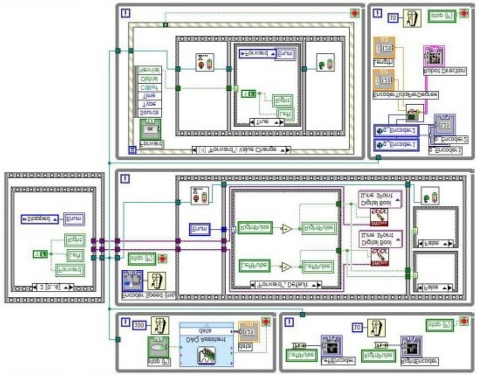
- Robot Specific Uses
  - PID loop refinement
  - Frame strain measurements
  - Electrical system diagnostics
  - Sensor Simulation
- Non-Robot Specific Uses
  - Dynamometer
  - Measuring ball velocities
  - Understanding FRC Electronics

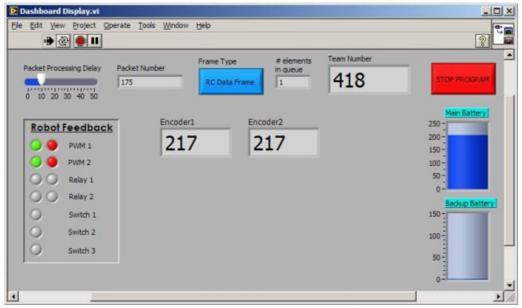


Sensor Simulation & Virtual Instruments

- Bench test your ***code*** with Virtual Instruments
  - Execute your code on the FRC Robot Controller
  - Simulate sensors using USB-DAQ
  - Visualize and Analyze robot responses with LabVIEW 8.0







LINX

LINX provides easy to use LabVIEW VIs for interacting with common embedded platforms like Arduino, chipKIT and myRIO. Use the built in sensor VIs to start getting data to your PC in seconds or use the peripheral VIs to access your devices digital I/O, analog I/O, SPI, I2C, UART, PWM and more.

Recommended Add-ons

NI-VISA

Optional Add-ons

LabVIEW Control Design and Simulation Module

LabVIEW MathScript RT Module

National Instruments VISA

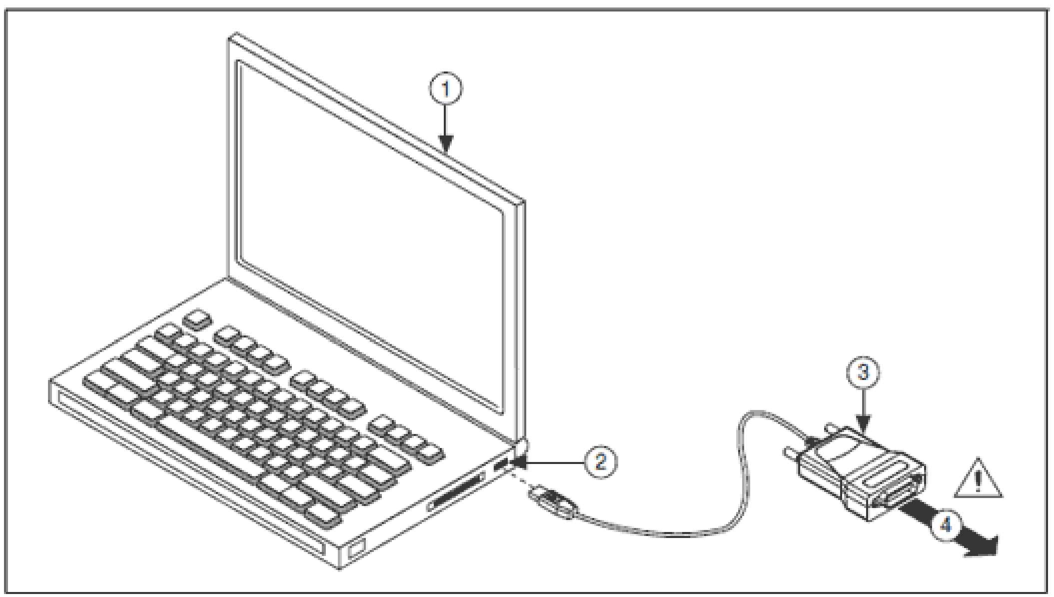
The Virtual Instrument Software Architecture (VISA) is a standard for configuring, programming, and troubleshooting instrumentation systems comprising GPIB, VXI, PXI, Serial, Ethernet, and/or USB interfaces. VISA provides the programming interface between the hardware and development environments such as LabVIEW, LabWindows/CVI, and Measurement Studio for Microsoft Visual Studio. NI-VISA is the National Instruments implementation of the VISA I/O standard. NI-VISA includes software libraries, interactive utilities such as NI I/O Trace and the VISA Interactive Control, and configuration programs through Measurement & Automation Explorer for all your development needs. NI-VISA is standard across the National Instruments product line. With NI-VISA, you can feel confident that your software development will not become obsolete as your instrumentation interface hardware needs evolve into the future.

Install Software and Drivers

The first step to getting started with NI products is to install the proper software and drivers for your application. Choose from the hardware platforms below for detailed, step-by-step instructions for installing LabVIEW software and NI device drivers.

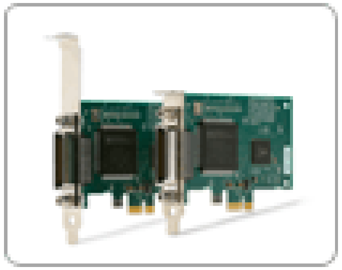
Instrument Control Instrument Control: Set Up Hardware

See the guide to installing LabVIEW, NI-VISA, and NI-488.2 to enable communication with any instrument via GPIB, USB, Ethernet/LAN, or Serial bus.



1 PC	2 USB Connector	3 GPIB-USB Interface	4 To GPIB Devices

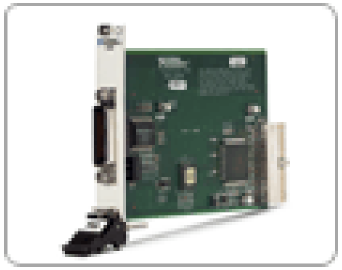
Figure 1. GPIB-USB Interface



PCI/PCI Express GPIB Controllers

GPIB-USB Controllers

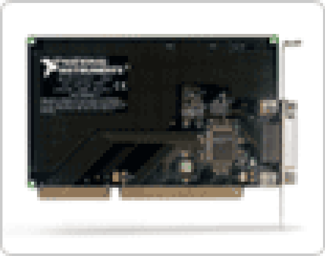
GPIB-ENET/100



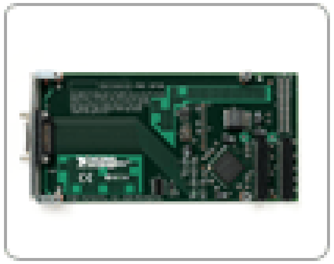
PXI GPIB and PXI-8232 Controllers

ExpressCard-GPIB Controllers

PCMCIA-GPIB Controllers



AT-GPIB/TNT (PnP) Controllers



PMC-GPIB Controllers