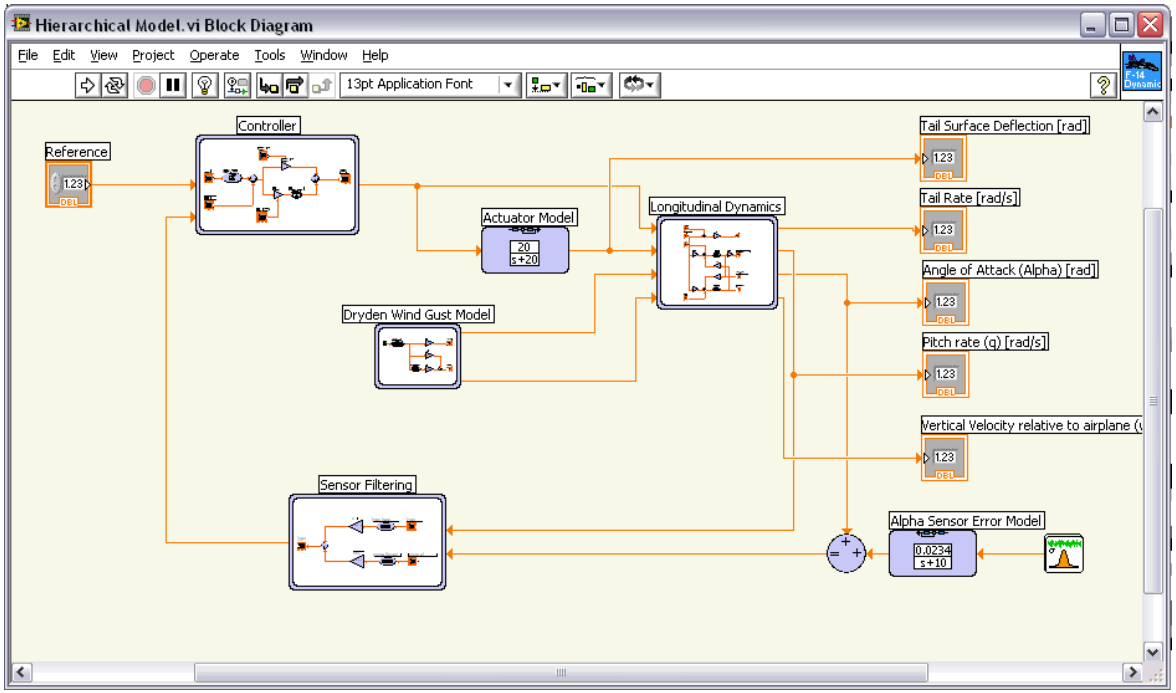
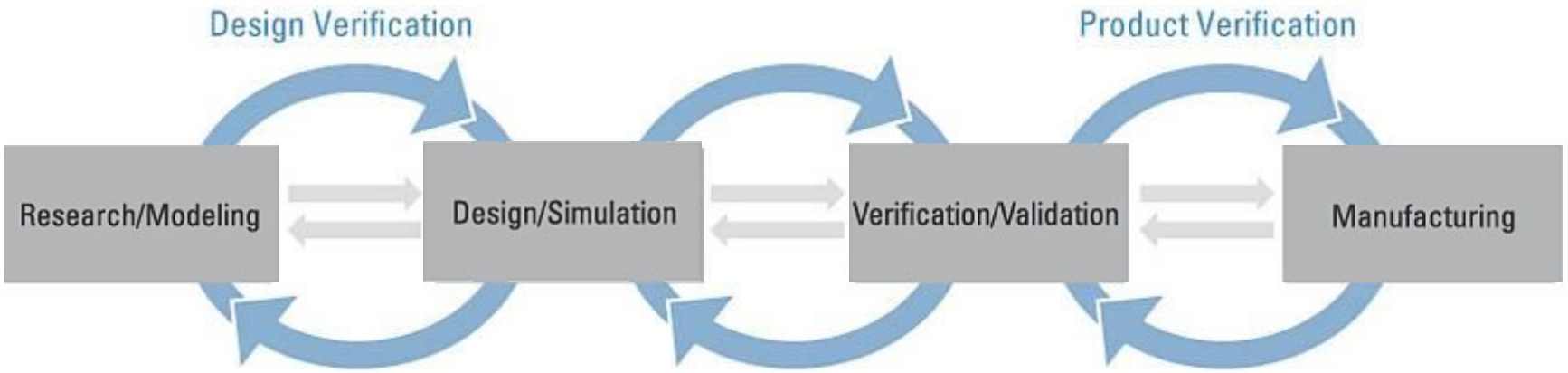
LabVIEW Control Design and Simulation

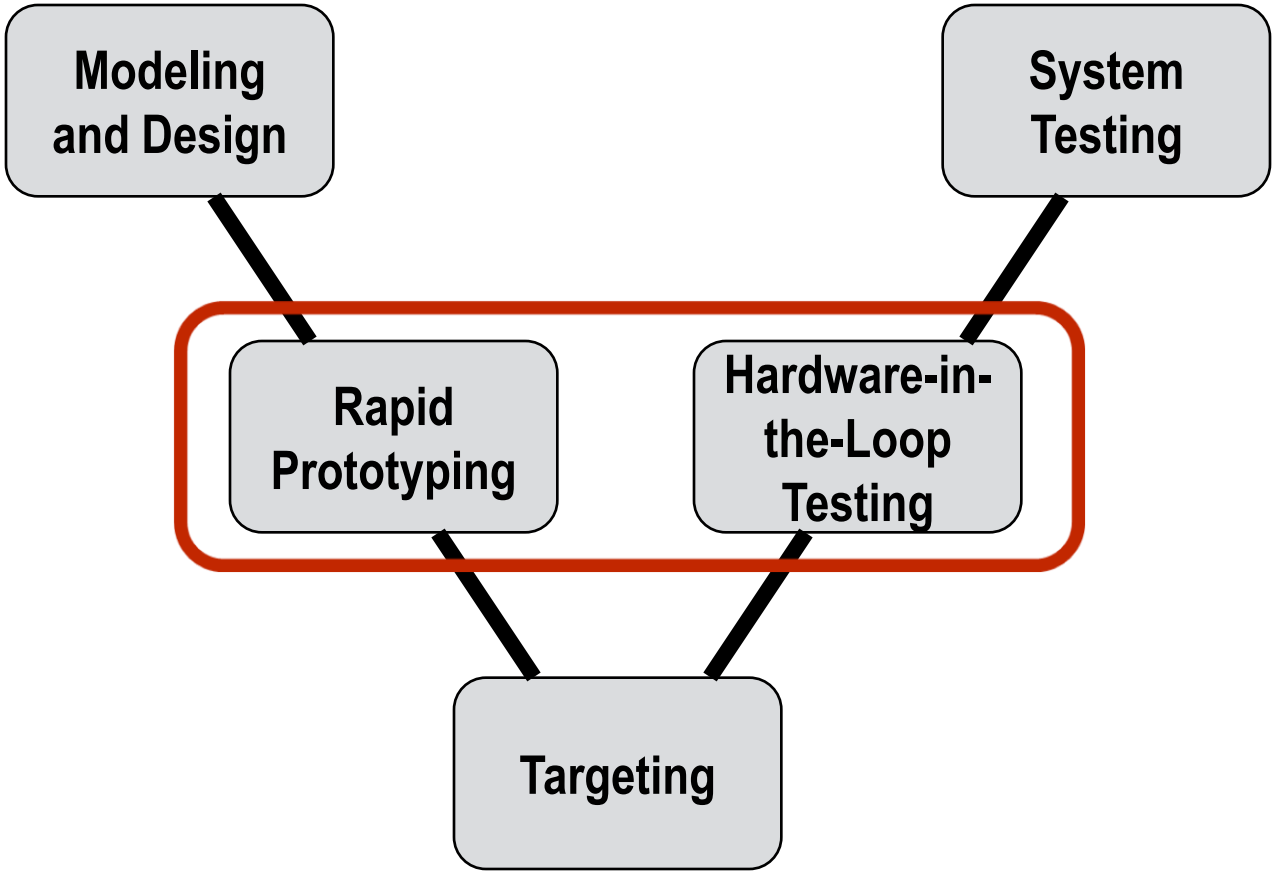


Jiří Keprt
Radim Štefan

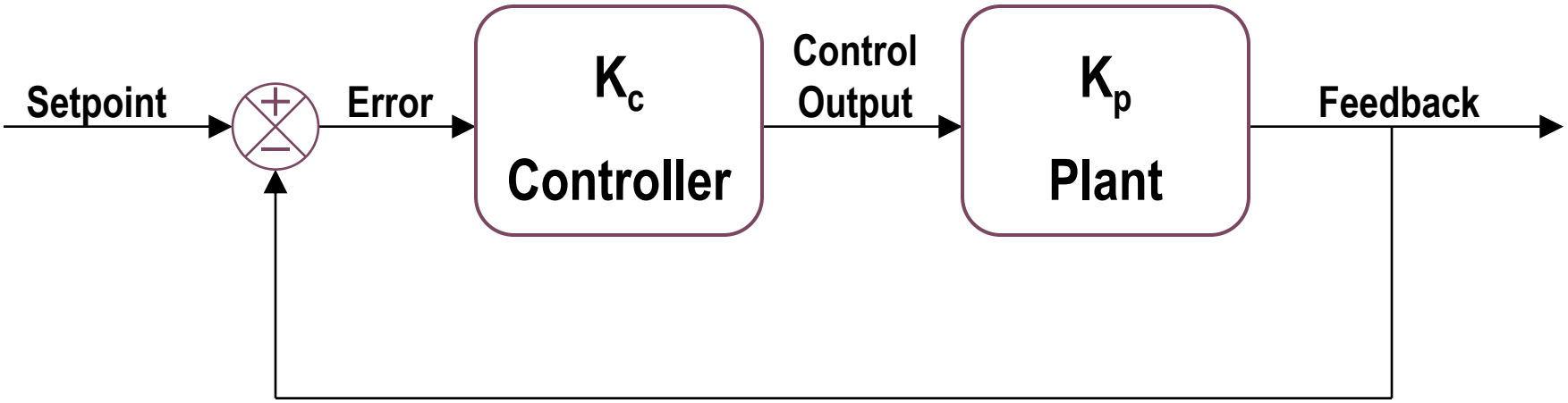
Control Design Process



V-diagram

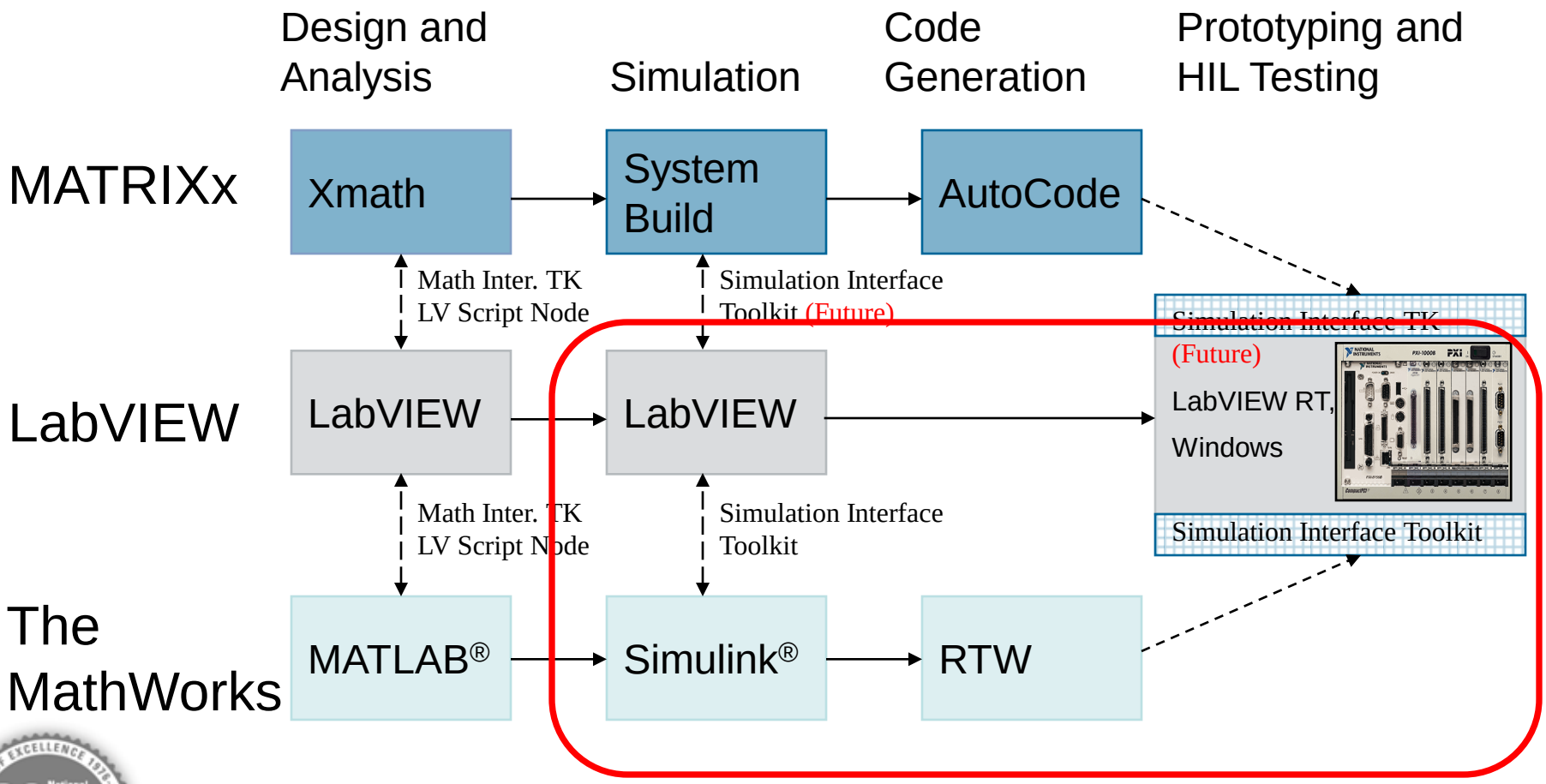

Modeling and Design
System Testing
Rapid Prototyping
Hardware-in-the-Loop Testing
Targeting

Modeling and Design



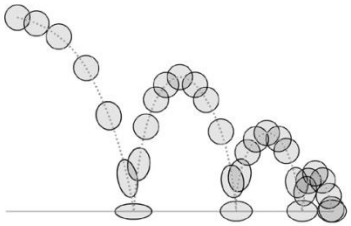
Modeling and design produce controller and plant models

Control Design Development Paths



Real-Time Workshop® is a registered trademark of The MathWorks, Inc. All other trademarks are the property of their respective owners.

Demo Bouncing Ball



- A rubber ball is thrown into the air with a velocity of 15 meters per second from a height of 10 m.
- Only gravitation is the force affecting the ball.
- The speed of the ball is decreased to 80 % when the ball hits the ground and the direction is reversed .
- The position of the ball is shown in the lower plot of the scope, and the velocity of the ball is shown in the upper plot.

Demo Bouncing Ball

- Differential equation of the model

$$m\ddot{x} = -mg$$

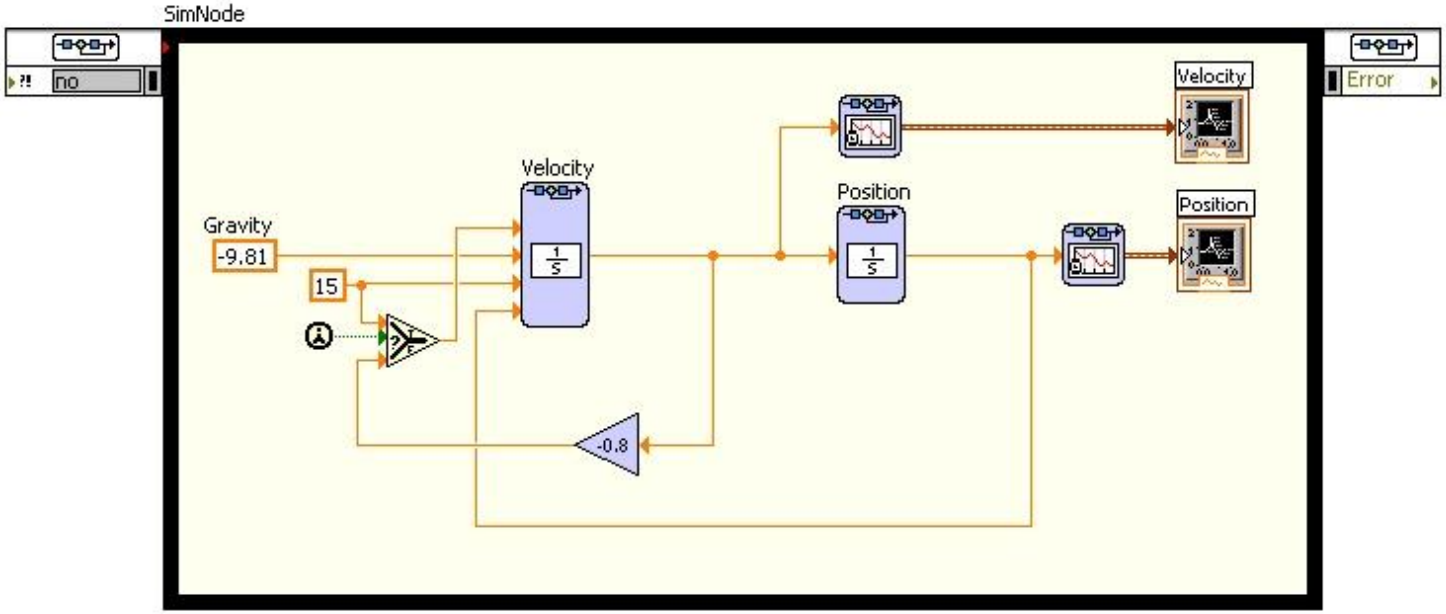
$$\ddot{x} = -g$$

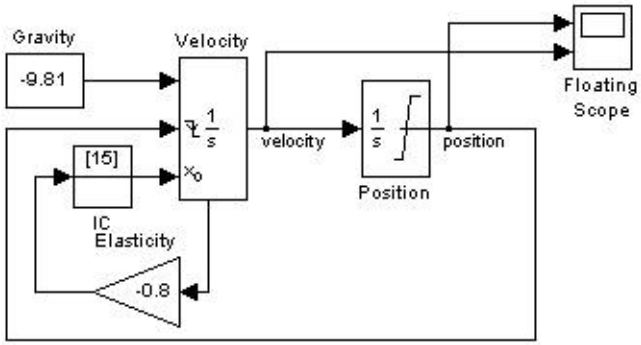
- Initial condition

$$\dot{x}_{[0]} = 15m/s$$

$$x_{[0]} = 10m$$

Demo Bouncing Ball

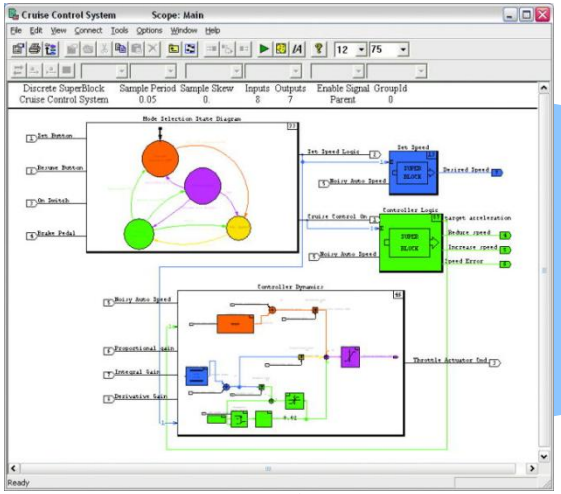


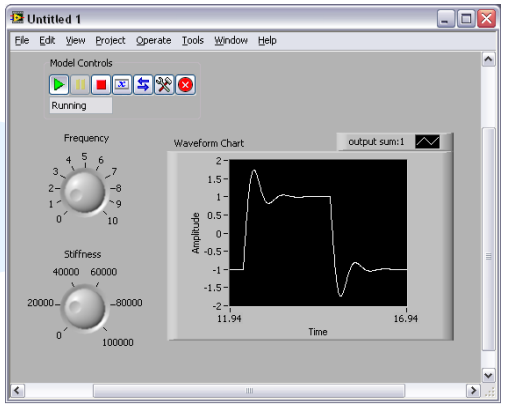


NI LabVIEW Simulation Interface Toolkit (SIT)

- Use the LabVIEW Simulation Interface Toolkit to:
  - Build powerful user interfaces for models developed in the Simulink® environment, and deploy them to real-time hardware with LabVIEW Real-Time*

    *Requires The MathWorks, Inc. Real-Time Workshop
  - Use Models from MatrixX

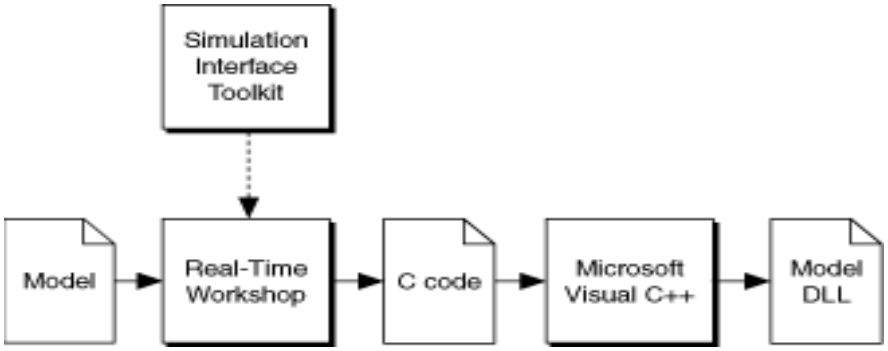


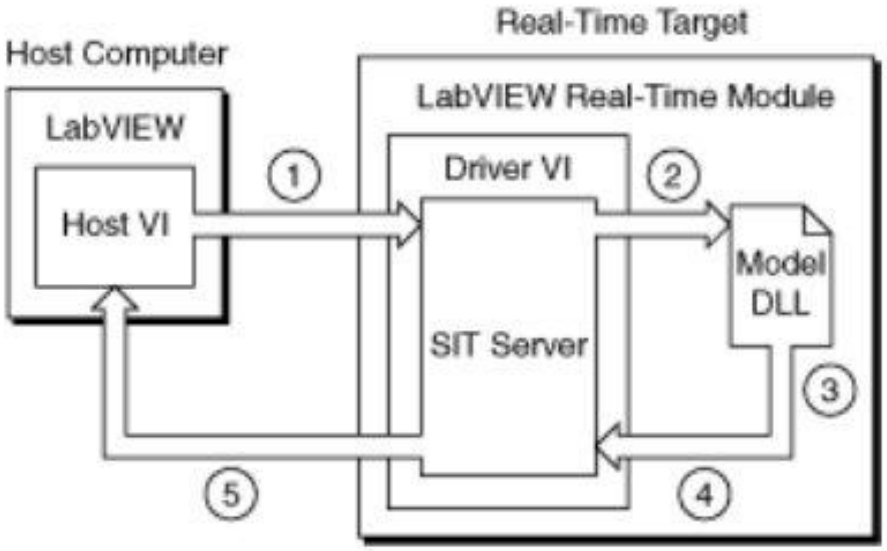


Real-Time Workshop® is a registered trademark of The MathWorks, Inc. All other trademarks are the property of their respective owners.

How it works?

Demo





Real-Time Workshop® is a registered trademark of The MathWorks, Inc. All other trademarks are the property of their respective owners.

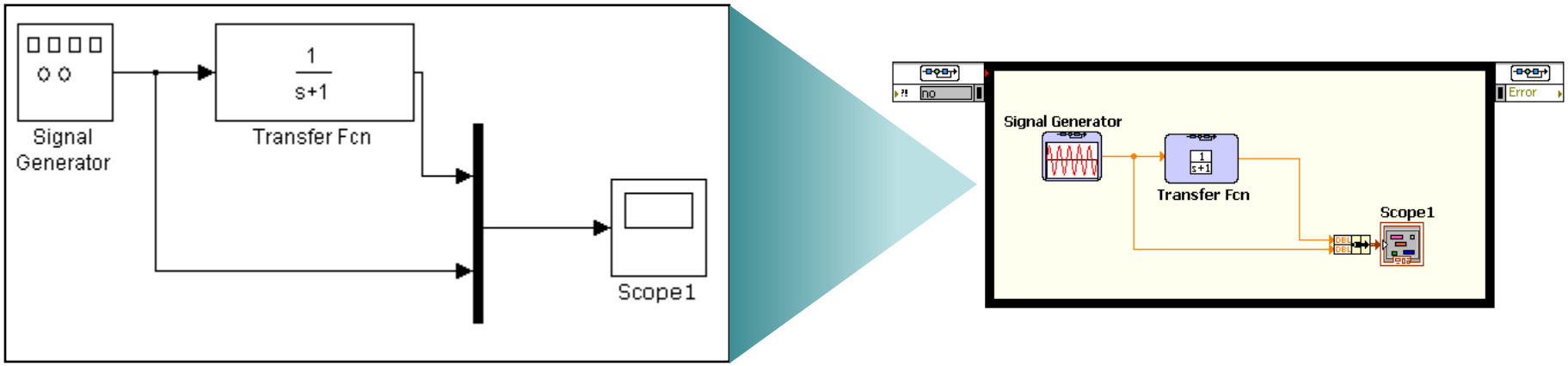
System Requirements

- The MathWorks MATLAB® 6.0 or later
- The MathWorks Simulink® 4.0 or later
- The MathWorks Real-Time Workshop® 4.0 or later
- Microsoft Visual C++ 5.0 or later
- Gcc WindRiver Compiler for VxWorks
- National Instruments LabVIEW 7.0 or later, Full or Professional Development Systems
- *National Instruments LabVIEW Real-Time Module*
- *NI-DAQ*

Simulink® is a registered trademark of The MathWorks, Inc. All other trademarks are the property of their respective owners.

Simulation Model Conversion

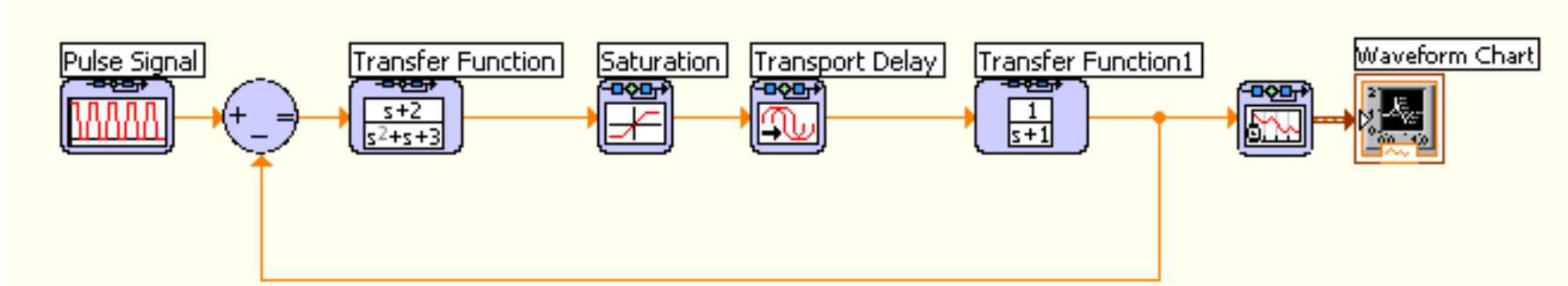
– Convert your plant and controller models developed in The MathWorks, Inc. Simulink® environment into LabVIEW Simulation Module code



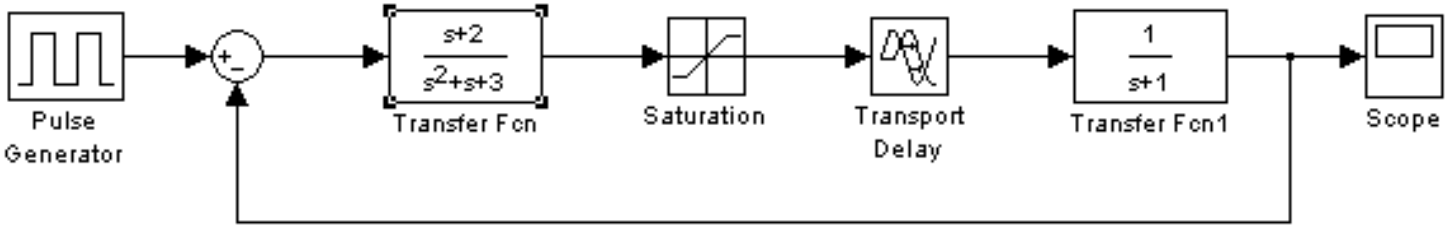
Simulink ® is a registered trademark of The MathWorks, Inc. All other trademarks are the property of their respective owners.

Little or No Learning Curve for The MathWorks, Inc. Simulink® Software Users

- LabVIEW Simulation Module

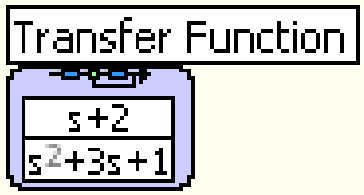


- The Simulink Software Environment

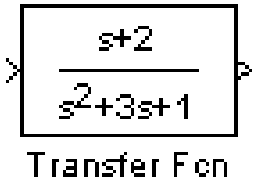


Simulink® is a registered trademark of The MathWorks, Inc. All other trademarks are the property of their respective owners.

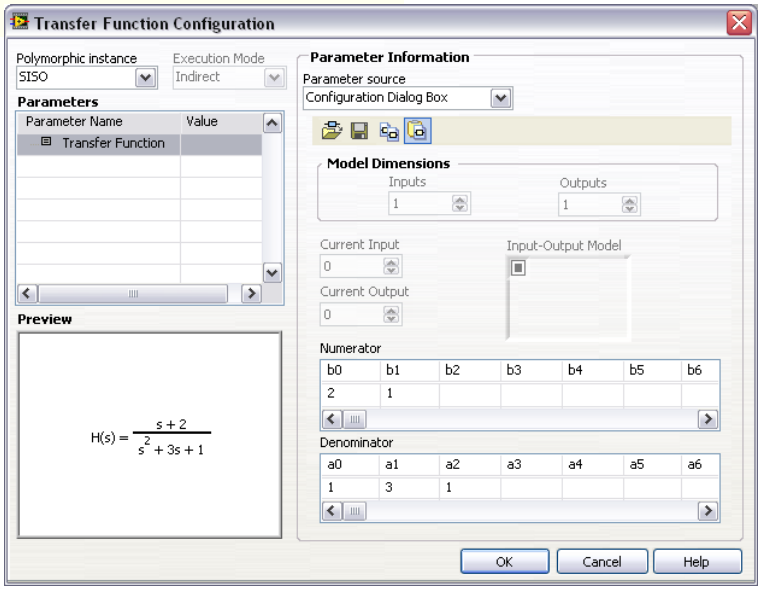
Little or No Learning Curve for The MathWorks, Inc. Simulink® Software Users

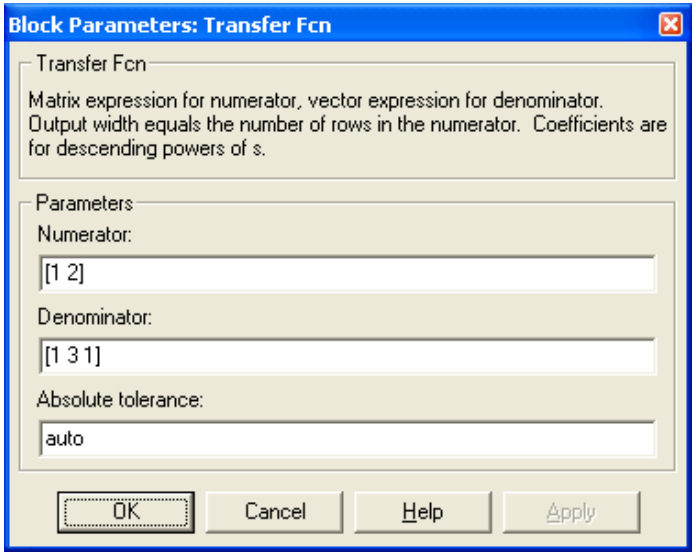


LabVIEW Simulation Module



The Simulink Software Environment

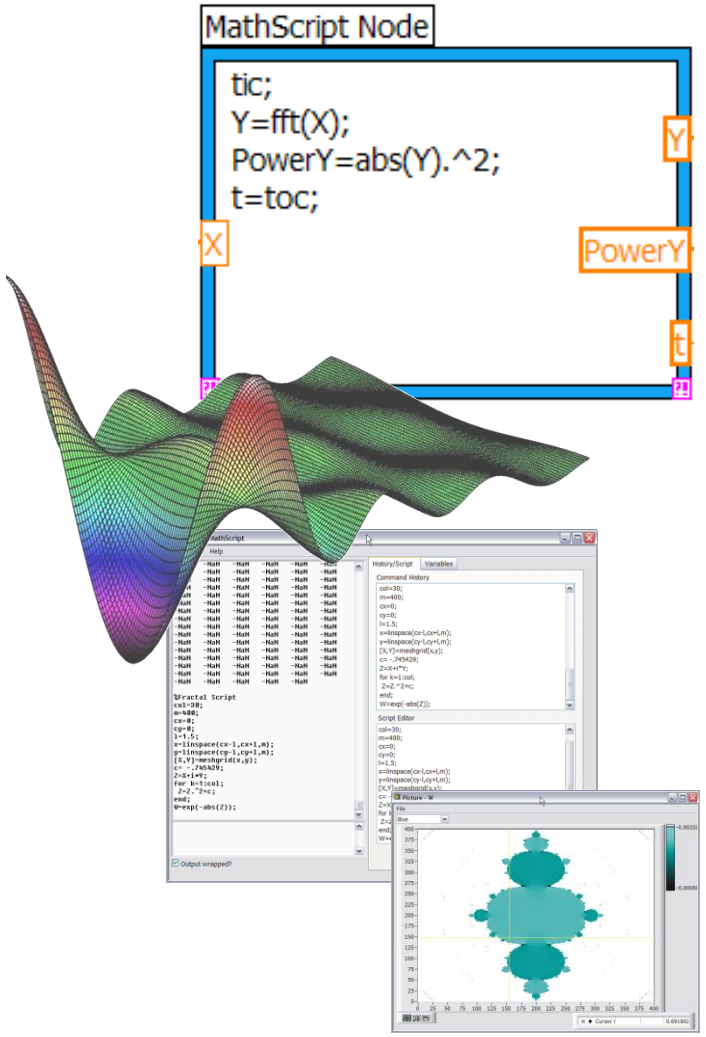




Simulink ® is a registered trademark of The MathWorks, Inc. All other trademarks are the property of their respective owners.

LabVIEW MathScript

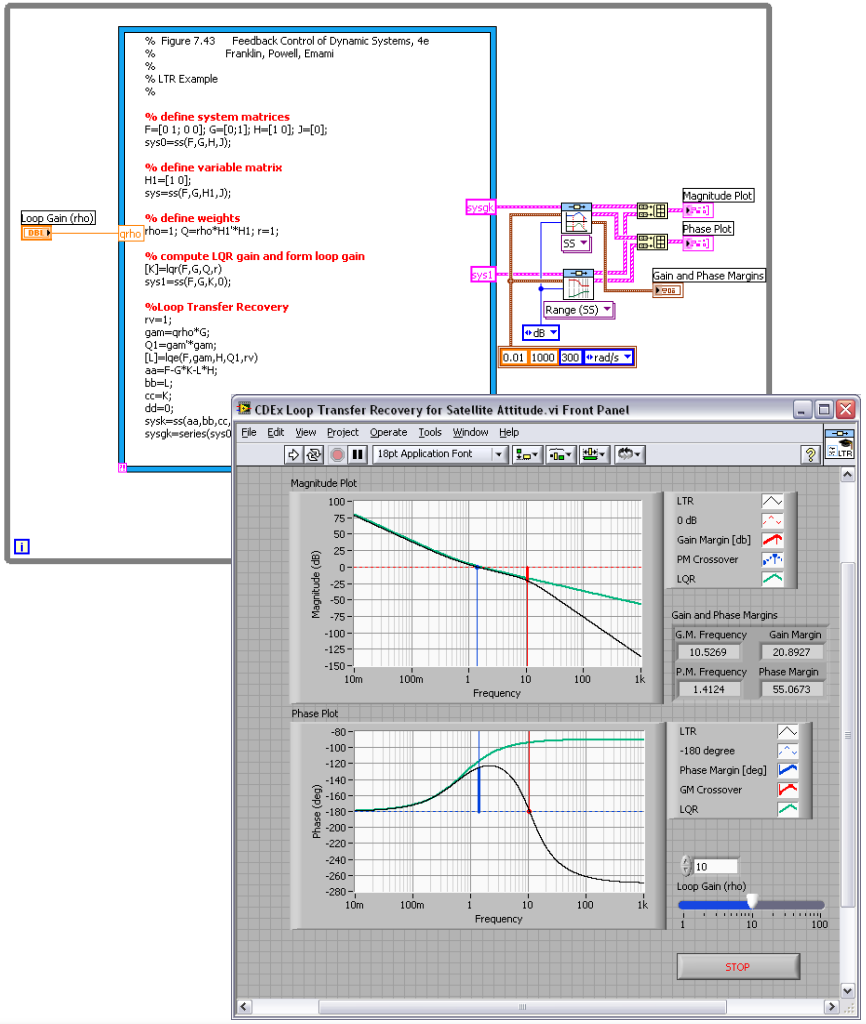
- **Powerful textual programming for signal processing, analysis, and math**
  - More than 650 built-in functions
  - Reuse many of your m-file scripts created with The MathWorks, Inc. MATLAB® software and others
  - Partially based on original math from NI MATRIXx
- **A native LabVIEW solution**
  - Interactive and programmatic interfaces
  - Does not require third-party software



MATLAB® is a registered trademark of The MathWorks, Inc. All other trademarks are the property of their respective owners.

Control Design MathScript Plug-In

- Use a MathScript node to combine your existing m-files with a flexible LabVIEW user interface
- Use MathScript text-based interactive environment for design and analysis
- Integrate controls work with built-in numerical math and signal processing functions in MathScript or native LabVIEW VIs and toolkits



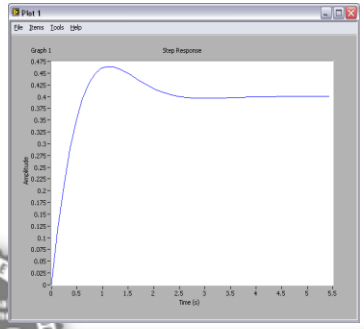
Little or No Learning Curve for Customers Familiar with The MathWorks Inc. MATLAB® Language Syntax

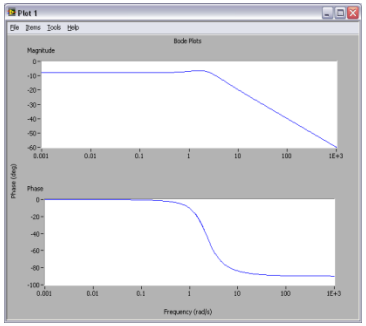
LabVIEW MathScript Syntax

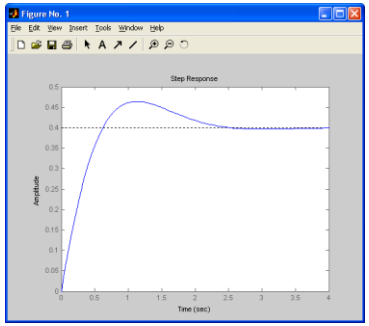
```
>>num = [1 2];

>>den = [1 3 5];

>>sys = tf(num,den);

>>step(sys)

>>bode(sys)

>>pole(sys)
ans =

          -1.5 + 1.6583i
          -1.5 - 1.6583i
```

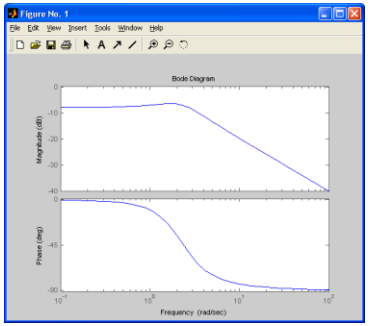
MATLAB ® syntax

```
>> num = [1 2];
>> den = [1 3 5];
>> sys = tf(num,den);
>> step(sys)
>> bode(sys)
>> pole(sys)

ans =

  -1.5000 + 1.6583i
  -1.5000 - 1.6583i
```



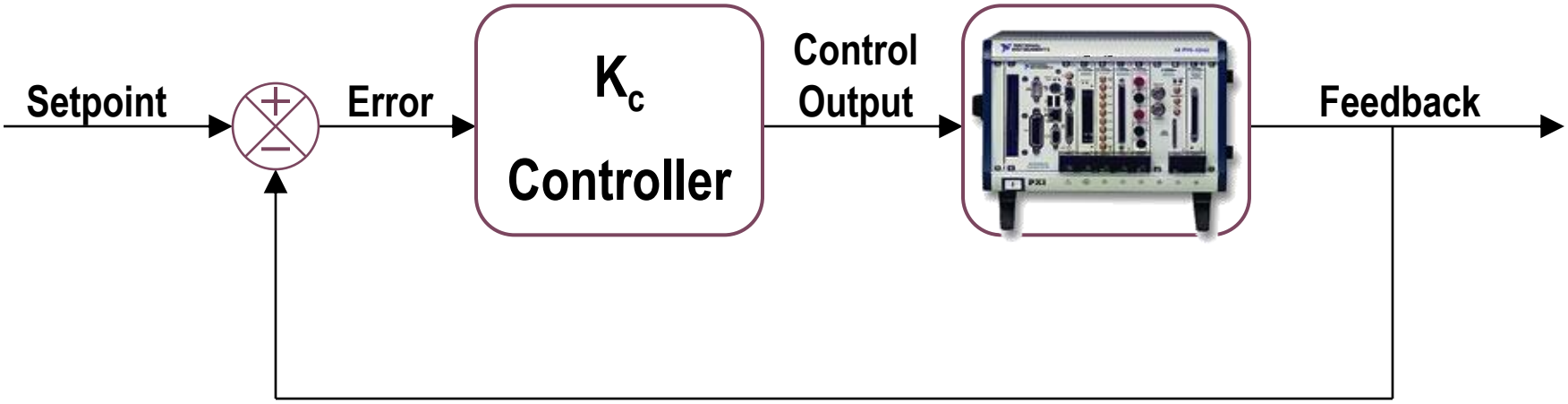






MATLAB® is a registered trademark of The MathWorks, Inc. All other trademarks are the property of their respective owners.

Hardware-in-the-Loop (HIL) Simulation



Testing production controller with simulated plant

LabVIEW for Design, Prototype, and Deploy

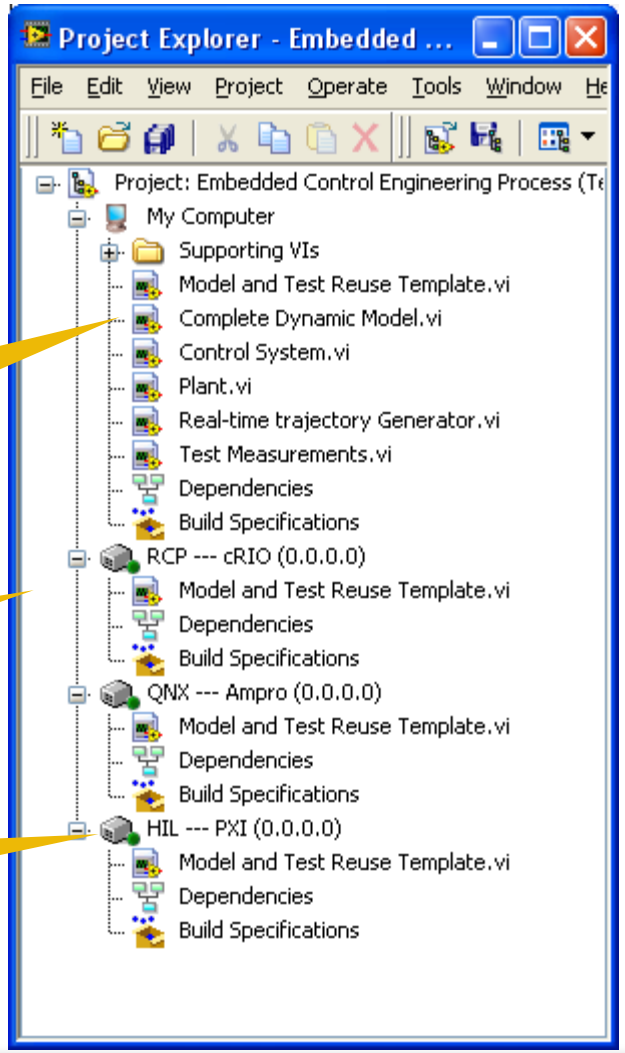
LabVIEW conditional compiling technology provides for:

- Model reuse
- Test reuse

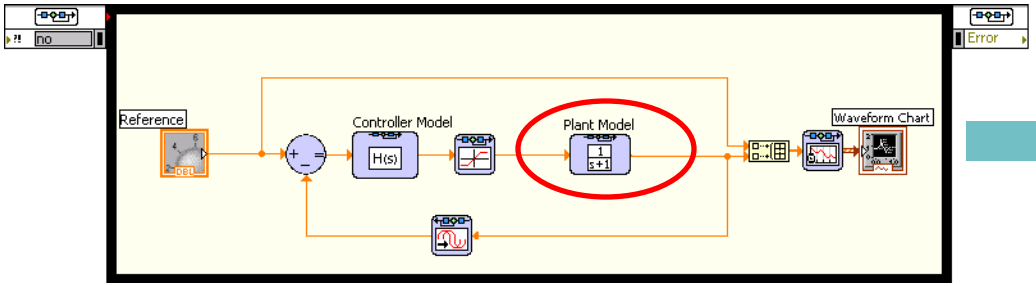
Design

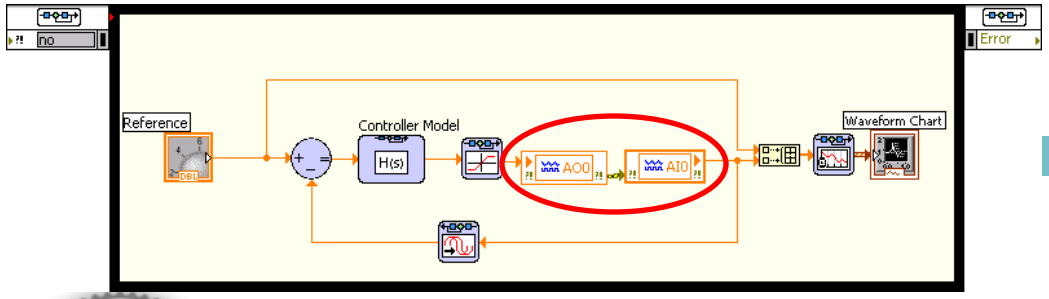
Prototype Target

HIL Target



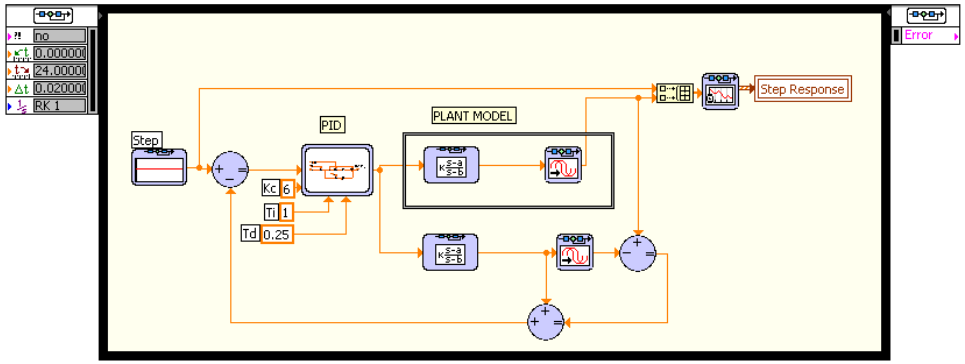
PAC Algorithm Deployment





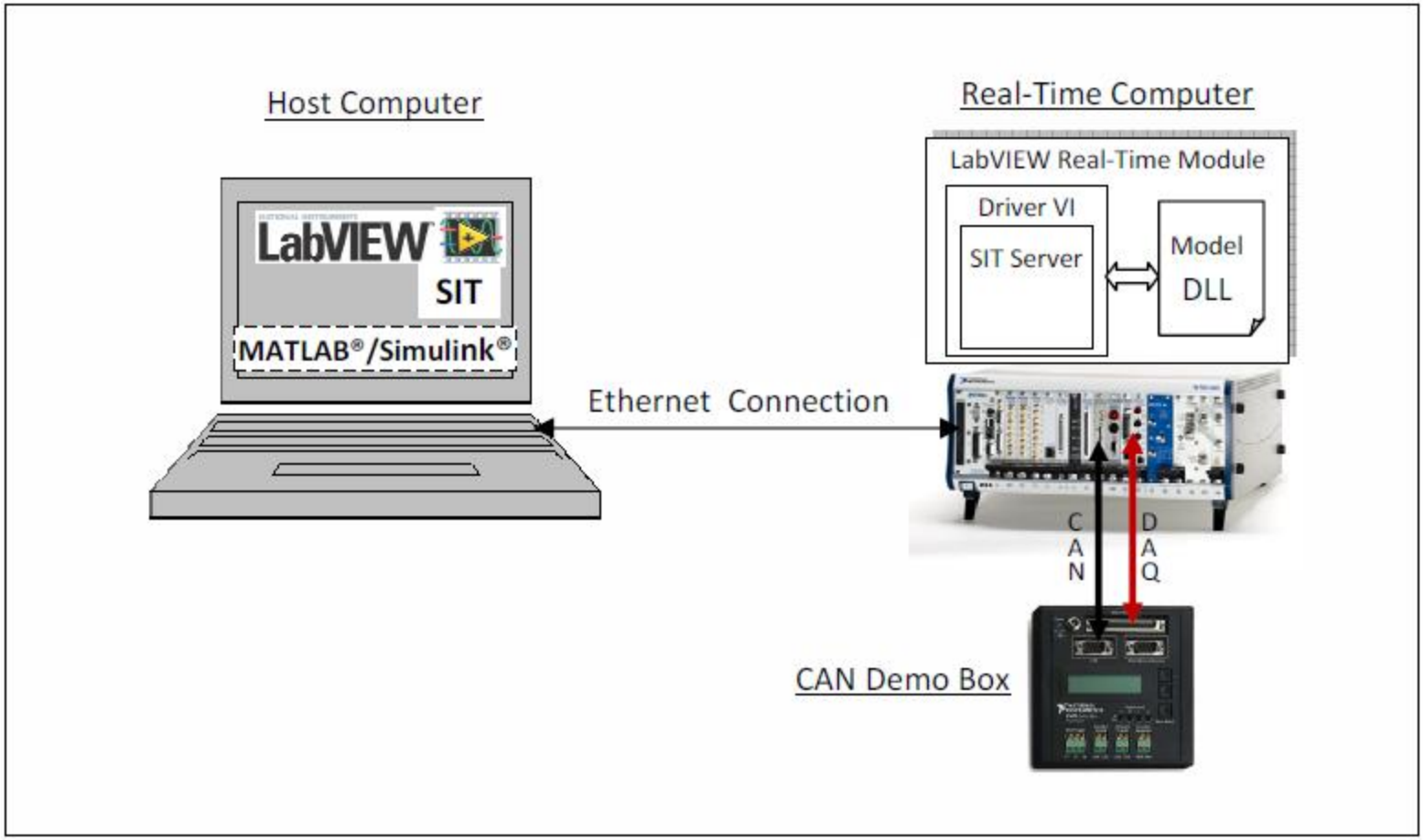
LabVIEW Control Design and Simulation Benefits

- ***Complete simulation and real-time implementation capability – stay in one environment from design to test to implementation***
- LabVIEW user interface to change and observe parameters as simulation or control system is running
- Use any LabVIEW VI or programming structure inside or outside of simulation loops:
  - Integrated design and simulation, batch simulation
  - DAQ, RIO, vision, or CAN for I/O and feedback
- Easily create parallel and multirate simulation or control loops



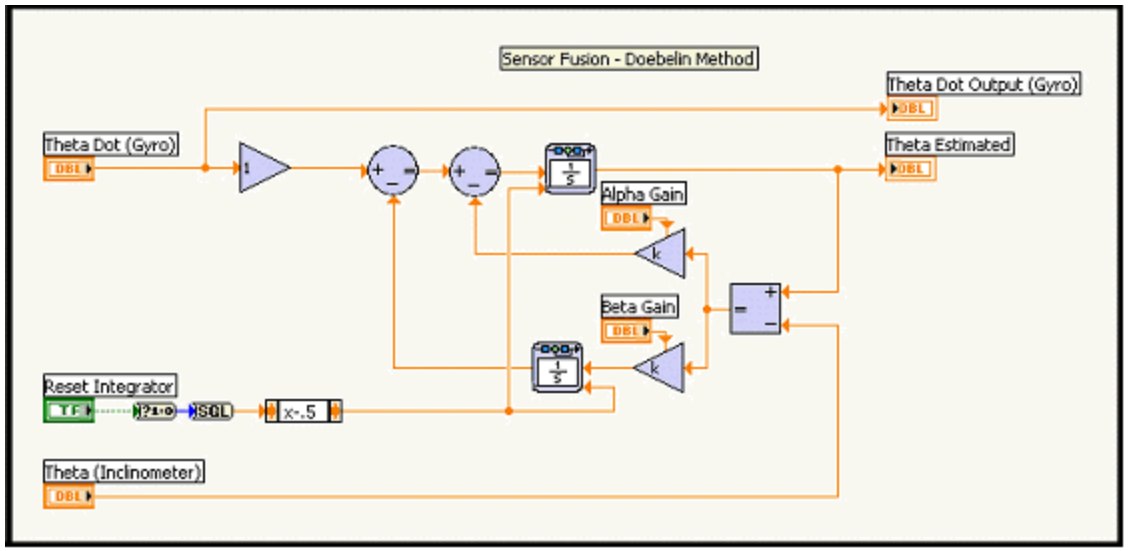
NATIONAL INSTRUMENTS

SIT / HIL Demo


Host Computer
Real-Time Computer
LabVIEW Real-Time Module
Driver VI
SIT Server
Model
DLL
LabVIEW
SIT
MATLAB®/Simulink®
Ethernet Connection
C
A
N
D
A
Q
CAN Demo Box

RPI Human Object Transporter

- Professor Kevin Craig, Rensselaer Polytechnic Institute (RPI)
- LabVIEW Simulation Module, CompactRIO, LabVIEW Real-Time, LabVIEW PDA
- Team of undergrads completed project in 4 months!



Děkujeme za pozornost