

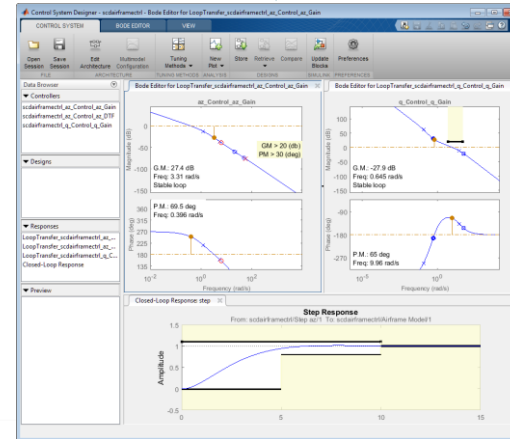
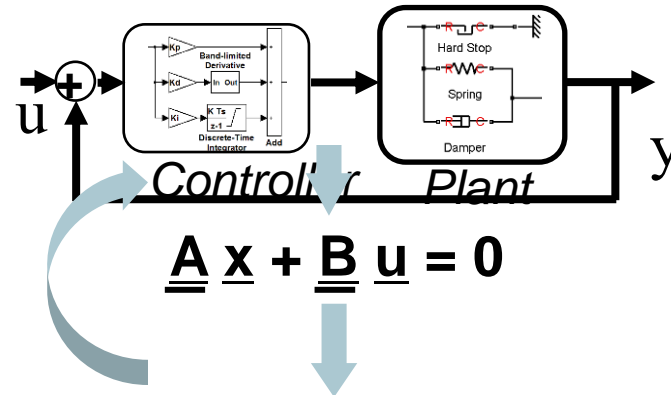
Simulation and Design of PID Controllers

Stefano Olivieri

MathWorks

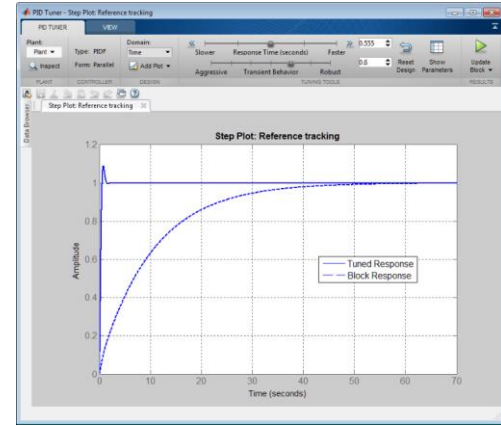
Introduction to Control Design

- Automatically tune arbitrary SISO and MIMO control systems modeled in Simulink
- Deploy PID Autotuning algorithm to embedded software
- Rapidly perform advanced linear analysis and control design for plants modeled in Simulink



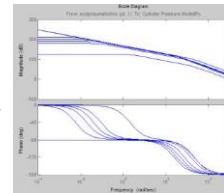
Key Features

- Automatic tuning of PID, gain-scheduled, and arbitrary SISO and MIMO control systems in Simulink
- PID autotuning algorithm deployable to embedded software
- Operating-point calculation (trimming) and model linearization
- Frequency response estimation from simulation data
- Batch linearization for varying parameters and operating points

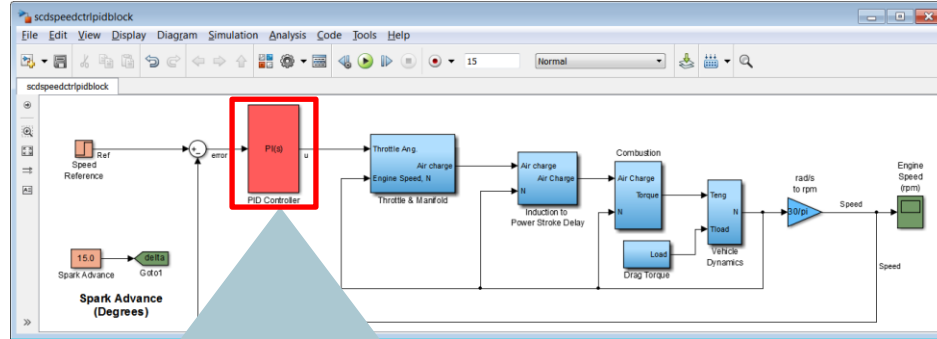
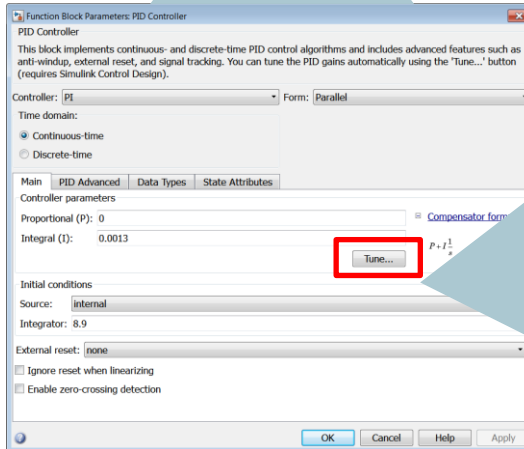


```

%% Gathering Simulation Snapshots
% To get operating point snapshots at various instants
% of the simulation use
op = findop('scdpneumaticlin',[0 10 20 30 40 50 60]);
%% The operating point is now ready for linearization.
% First specify the input and output points using the commands:
io(1) = linio('scdpneumaticlin/x',1,'in');
io(2) = linio('scdpneumaticlin/Cylinder Pressure Model',1,'out');
%% Linearize the model and plot the Bode plot for each condition
% to see the variability in the linearizations.
sys = linearize('scdpneumaticlin',op,io);
bode(sys)
    
```



Design and Tuning of PID Controllers

Function Block Parameters: PID Controller

This block implements continuous- and discrete-time PID control algorithms and includes advanced features such as anti-windup, external reset, and signal tracking. You can tune the PID gains automatically using the "Tune..." button (requires Simulink Control Design).

Controller: **PI** Form: **Parallel**

Time domain:
 Continuous-time
 Discrete-time

Main **PID Advanced** Data Types State Attributes

Controller parameters
 Proportional (P): 0
 Integral (I): 0.0013 $P + I \frac{1}{s}$

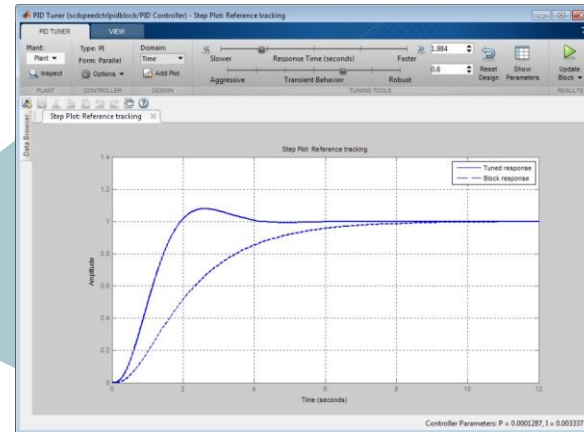
Tune...

Initial conditions
 Source: **internal**
 Integrator: 8.9

External reset: **none**

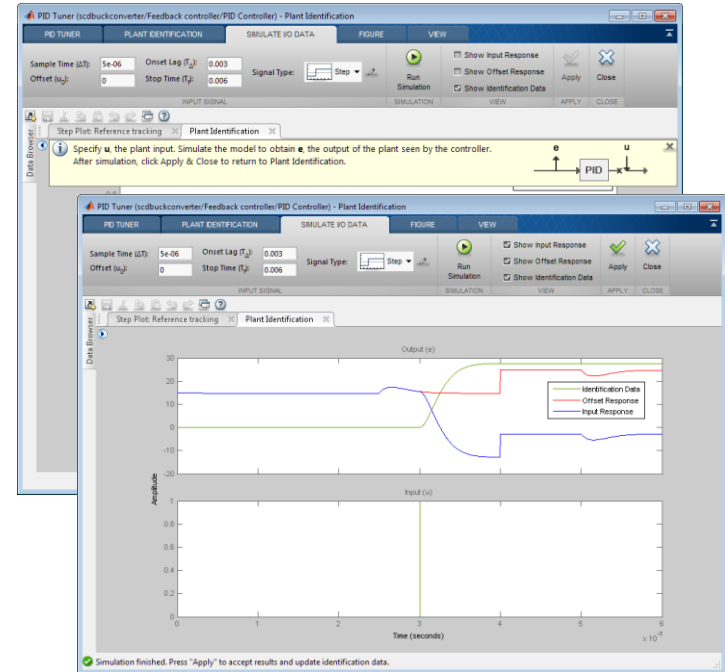
Ignore reset when linearizing
 Enable zero-crossing detection

OK Cancel Help Apply



System Identification Integrated into PID Tuner

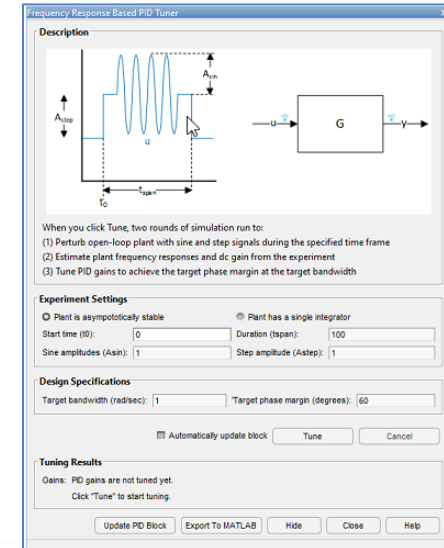
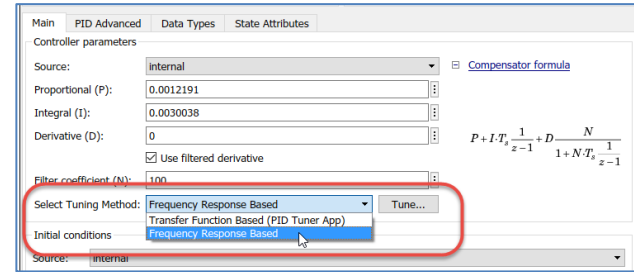
- Tune PID Controllers for Simulink models with discontinuities such as PWM and Stateflow logic
- Compute plant transfer function from simulation input-output data when exact linearization fails
- Inject a step or an impulse at the plant input
- Interactively or automatically fit the transfer function to simulation input-output data



PID Autotuning

Automatically tune PID controller gains for models with plants that do not linearize

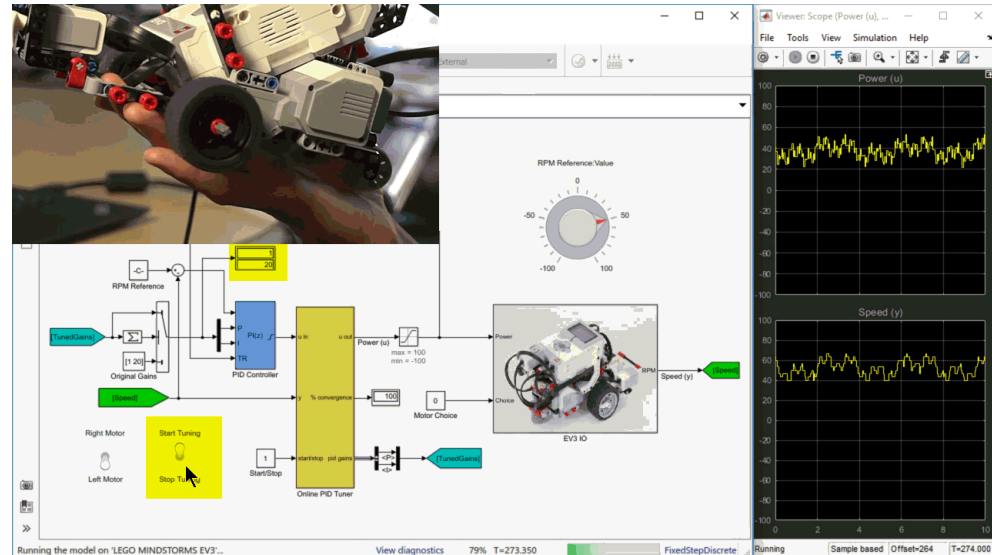
- Frequency-response based PID Tuner simulates the model to estimate its frequency response at a few frequencies near control bandwidth
- Controller gains are automatically computed from this frequency response
- The method can be used for models that do not linearize, for example, power electronics systems



PID Autotuning

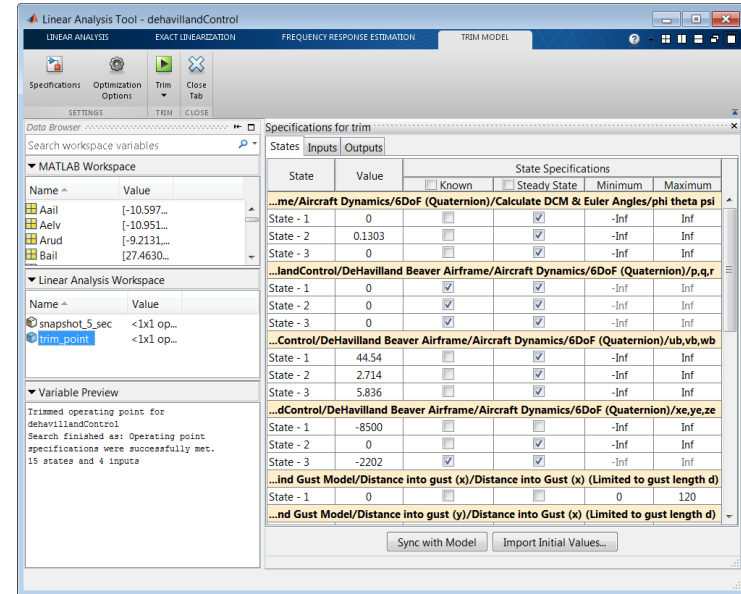
Deploy PID autotuning algorithm to embedded software

- Use Online PID Tuner block to generate autotuning code and deploy to embedded software
- PID autotuning is model-free, no a priori model of the plant is required
- PID autotuning generated code can be used completely independently from Simulink or through External Mode

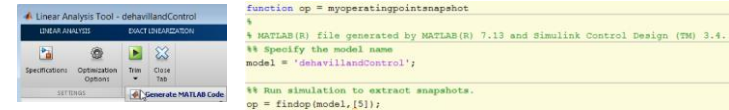


Nonintrusive Trimming of Simulink Models

- Find trim points using optimization methods
- Trim at specific times or events during simulation
- Automatically generate MATLAB code from the Graphical User Interface



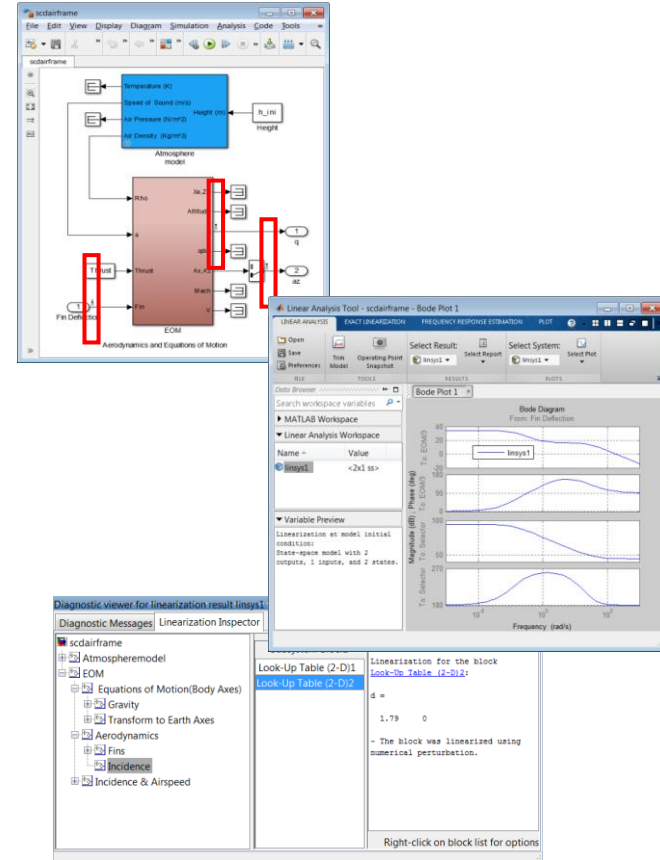
State	Value	State Specifications			
		Known	Steady State	Minimum	Maximum
...me/Aircraft Dynamics/6DoF (Quaternion)/Calculate DCM & Euler Angles/phi theta psi					
State - 1	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	-Inf	Inf
State - 2	0.1303	<input type="checkbox"/>	<input checked="" type="checkbox"/>	-Inf	Inf
State - 3	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	-Inf	Inf
...landControl/DeHavilland Beaver Airframe/Aircraft Dynamics/6DoF (Quaternion)/p,q,r					
State - 1	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	-Inf	Inf
State - 2	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	-Inf	Inf
State - 3	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	-Inf	Inf
...Control/DeHavilland Beaver Airframe/Aircraft Dynamics/6DoF (Quaternion)/ub,yb,wb					
State - 1	44.54	<input type="checkbox"/>	<input checked="" type="checkbox"/>	-Inf	Inf
State - 2	2.714	<input type="checkbox"/>	<input checked="" type="checkbox"/>	-Inf	Inf
State - 3	5.836	<input type="checkbox"/>	<input checked="" type="checkbox"/>	-Inf	Inf
...dControl/DeHavilland Beaver Airframe/Aircraft Dynamics/6DoF (Quaternion)/xe,ye,ze					
State - 1	-8500	<input type="checkbox"/>	<input checked="" type="checkbox"/>	-Inf	Inf
State - 2	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	-Inf	Inf
State - 3	-2202	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	-Inf	Inf
...ind Gust Model/Distance into gust (x)/Distance into Gust (x) (Limited to gust length d)					
State - 1	0	<input type="checkbox"/>	<input type="checkbox"/>	0	120
...nd Gust Model/Distance into gust (y)/Distance into Gust (x) (Limited to gust length d)					



```
function op = myoperatingpointnapahot
%
% MATLAB (R) file generated by MATLAB (R) 7.13 and Simulink Control Design (TM) 3.4.
%% Specify the model name
model = 'dehavillandControl';
%% Run simulation to extract snapshots.
op = findop(model,[1]);
```


Nonintrusive Linearization of Simulink Models

- Linearize without having to modify model structure
- Linearize whole model, portion of model, single block, or subsystem
- Specify linearization behavior of any number of blocks in the model
 - Linearize models containing discontinuities or event-based logic
 - Compute uncertain linear models for use with Robust Control Toolbox
- Use linearization Inspector to view and plot linearization results



Simulation-Based Computation of a Simulink Model Frequency Response

- Compute frequency response for models with strong discontinuities or event-based dynamics
- Verify results of a linearization
- Study the effects of excitation signal amplitude on nonlinear system's gain and phase characteristics

