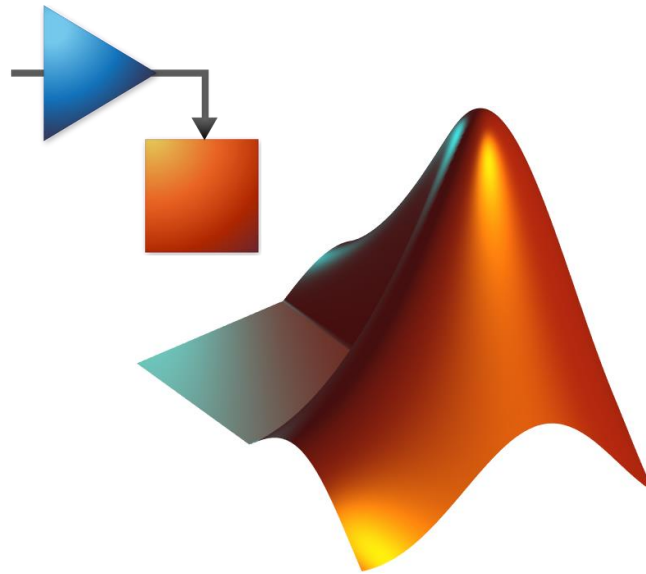


MATLAB / Simulink Lab Course

Simulink Control Design



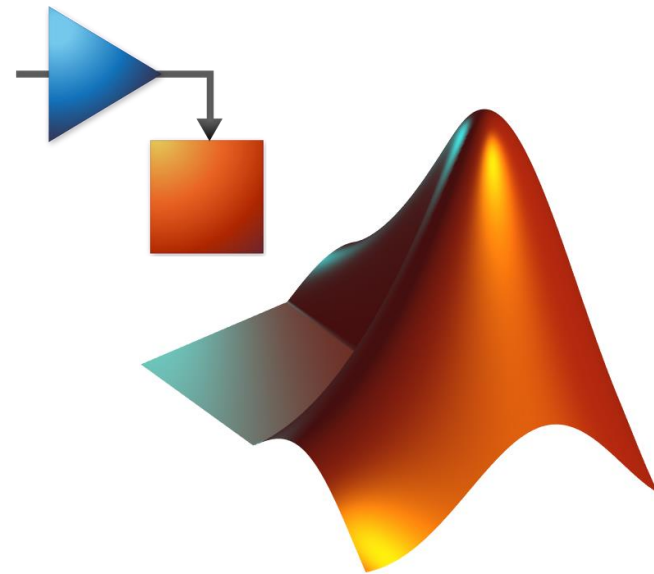
Objectives & Preparation “Simulink Control Design”

- Which MathWorks products are covered?
 - ⇒ Simulink Control Design
- What skills are learnt?
 - ⇒ Search for operating points of Simulink models
 - ⇒ Linearize the system around operating points
 - ⇒ Estimate the frequency response of the system
 - ⇒ Tune controllers with the Simulink Control Design
 - ⇒ Verify the model and the controlled closed-loop system
- How to prepare for the session?
 - ⇒ Simulink Control Design documentation:
 - <https://de.mathworks.com/help/slcontrol/>
 - ⇒ Tutorials for each section of this session:
 - <https://de.mathworks.com/help/slcontrol/getting-started-with-simulink-control-design.html>

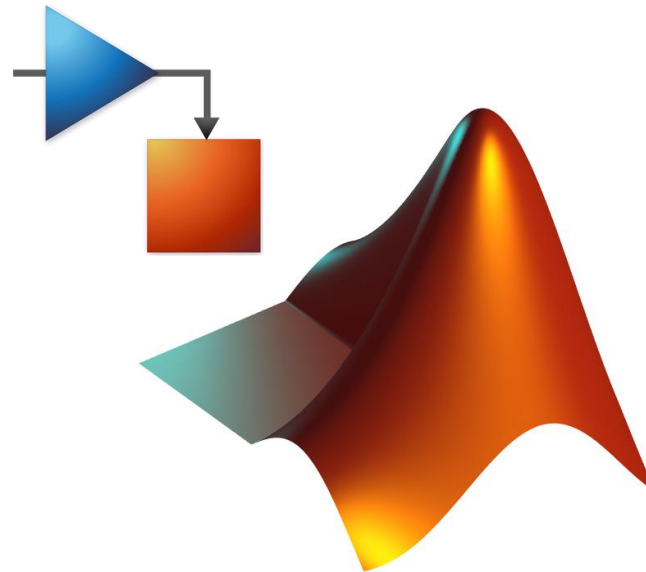


Outline

1. Introduction
2. Operating Points
3. Linearization
4. Frequency Response Estimation
5. Control Design
6. Model Verification



1. Introduction




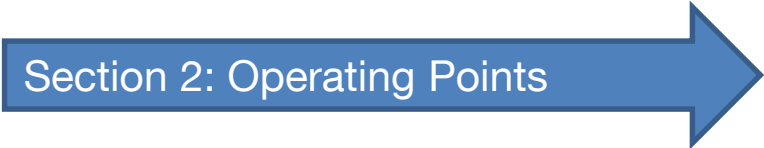
Introduction


- Similar to MATLAB's Control System Toolbox, **Simulink Control Design** is a useful tool for **analysis and design of dynamic (control) systems**, which are modeled in Simulink.
- While the **Control System Toolbox** describes **linear** systems only, **Simulink Control Design** is additionally capable of working with **nonlinear systems**.
- The functionalities of the Simulink Control Design toolbox presented here include:
 - Nonintrusive computation of operating points (trimming) of Simulink models
 - Exact linearization of nonlinear Simulink models
 - Simulation-based computation of frequency responses
 - Graphical and automated tuning of PID controllers and arbitrary controller layouts.
 - Numerical optimization of compensators to meet time-domain or frequency-domain requirements.
- Other toolboxes that are helpful for control design using Simulink include the Simulink Design Optimization (SDO) toolbox, the Simulink Design Verifier and the Robust Control Toolbox. They are not covered by this lab course. Visit the MATLAB lecture or refer to the MATLAB help to learn more!


Introduction


- The following list of steps puts the features of the toolbox and the sections of this course into the context of actual control system development:
 1. Build a Simulink model of the plant
(Suppose this model is nonlinear.)

Session: Simulink Fundamentals
 2. Determine (relevant) operating points of the plant

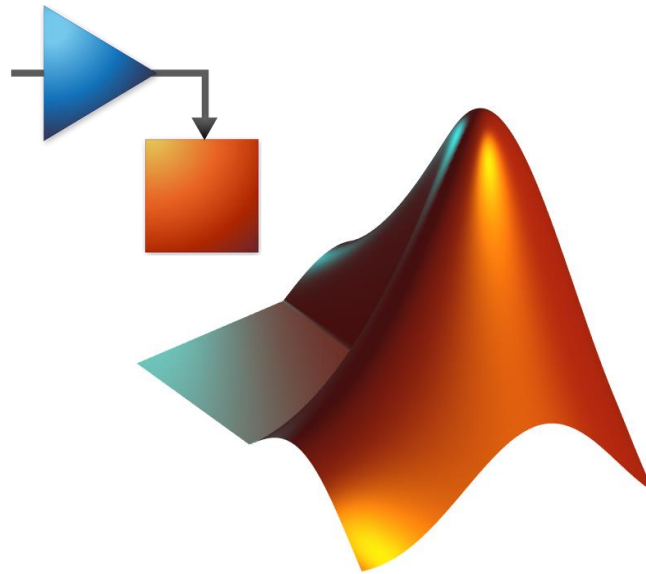
Section 2: Operating Points
 3. Linearize the plant at those operating points
(and possibly verify the linearization)

Sections 3 & 4: Linearization, FRE
 4. Design a linear controller, based on the linearized plant model

Section 5: Control Design
 5. Verify that the linear controller performs as desired with the nonlinear plant

Section 6: Model Verification

2. Operating Points

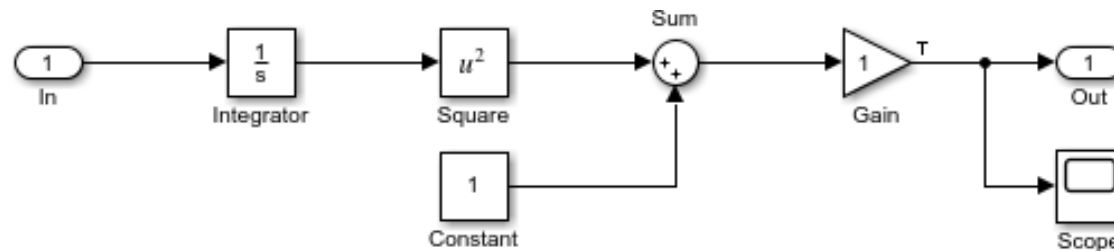


Operating Points

- An operating point is the state of a dynamic system at a specific point in time.
- A steady-state operating point, also called equilibrium or trim condition, is an operating point at which (some) state variables do not change over time.
- After an operating point has been found, the system can be linearized (see section 3), so that linear control theory can be applied.
- Simulink itself provides a function that finds operating points, named trim. The Simulink Control Design toolbox, however, brings substantial advantages:
 - A Graphical User Interface (GUI)
 - Multiple optimization methods
 - Constraints on state, input, and output variables using upper and lower bounds
 - Steady-state operating points for models with discrete states
 - Model reference support

Operating Points

- The operating point of a Simulink model consists of the **model initial states** and **root-level input signals**. In the case of the following model, this is
 - The input In1
 - The state of the Integrator block



- States are introduced by blocks like integrators, transfer function-, state space-, PID-, or zero-pole-gain-blocks (or their discrete-time counterparts).
- Inputs are represented by inports.
- Outputs are represented by outports or by “Trim Output Constraints”: any signal in a model can be specified a Trim Output Constraint by *Right Click* → *Linear Analysis Points* → *Trim Output Constraint*. The signal arrow is accompanied by the letter “T”.

T Trim Output Constraint

Operating Points

- You can find steady-state operating points from user specifications or after specific simulation times. While the first method is optimization-based (“trimming”), the latter method is simulation-based and also referred to as “**simulation snapshot**.”

Optimization-based search

Use this approach

- when you know some of the **operating point states** and **model input or output signal levels**.

This approach produces poor results:

- When you specify **initial guesses** far away from the desired operating point.
- When you specify incompatible input, output, or state constraints at equilibrium.

Simulation-based search

Use this approach

- when the **simulation time is sufficiently short** for the model to reach steady state.

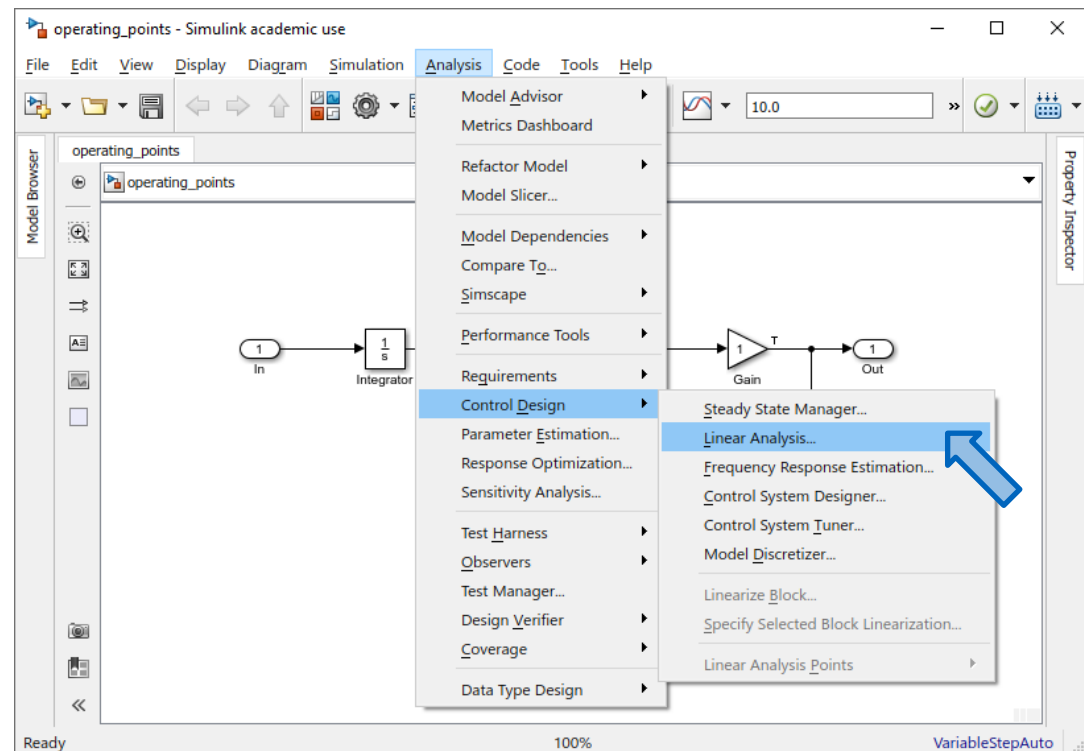
This approach produces poor results:

- If the model takes too long to reach steady state.
- When you specify **initial conditions** that do not cause the model to reach true equilibrium.

Both approaches can be combined: First a simulation snapshot, then an optimization.

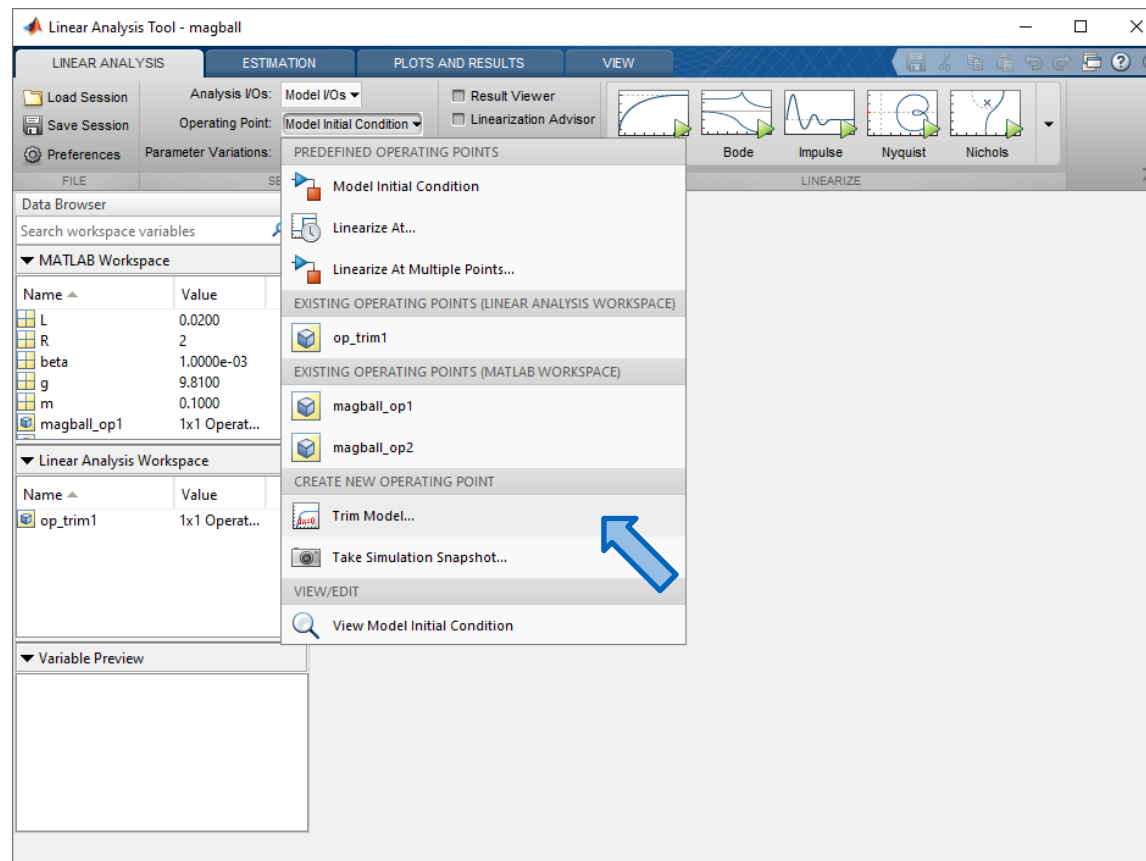
Operating Points

- Three different search methods for steady-state operating points are presented here:
 1. Optimization-based search for steady-state operating points from state specifications (trimming)
 2. Simulation-based search for steady-state operating points (simulation snapshot)
 3. Initialization of optimization-based search using simulation snapshots
- All of them – and also many other functionalities presented in the following sections – can be accessed through the Linear Analysis Tool.



Operating Points – Optimization-based Search

- On the Linear Analysis Tool GUI, choose “Trim Model...” from the “Operating Point” dropdown menu to open a dialog box for operating point search.



Operating Points – Optimization-based Search

- By default, the list of states is shown. You can specify the system states and trim.

Select "Known" to fix the state value during trimming.

Specify whether a state shall be steady.

Specify minimum and/or maximum values.

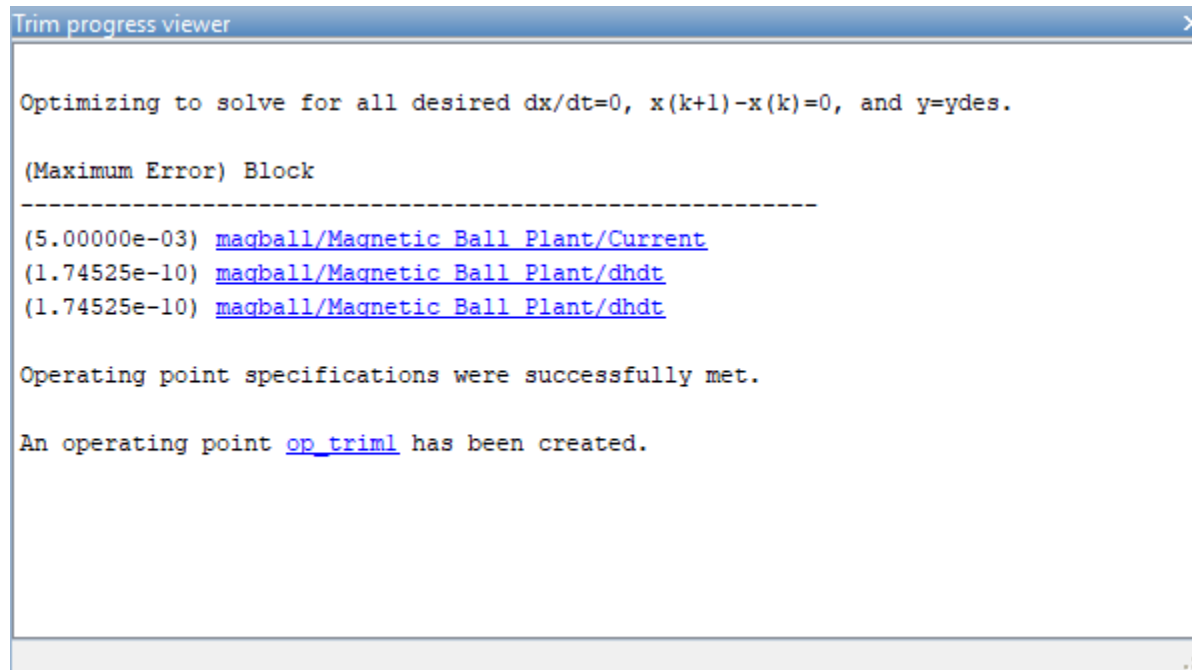
Once everything is specified, click here to start trimming.

State	Value	<input type="checkbox"/> Known	<input checked="" type="checkbox"/> Steady State	Minimum	Maximum	dx Minimum	dx Maximum
magball/Controller/PID Controller/Filter/Cont. Filter/Filter							
State - 1	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	-Inf	Inf	-Inf	Inf
magball/Controller/PID Controller/Integrator/Continuous/Integrator							
State - 1	14.0071	<input type="checkbox"/>	<input checked="" type="checkbox"/>	-Inf	Inf	-Inf	Inf
magball/Magnetic Ball Plant/Current							
State - 1	7.0036	<input type="checkbox"/>	<input checked="" type="checkbox"/>	0	Inf	-Inf	Inf
magball/Magnetic Ball Plant/dhdt							
State - 1	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	-Inf	Inf	-Inf	Inf
magball/Magnetic Ball Plant/height							
State - 1	0.05	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	-Inf	Inf	-Inf	Inf

- Alternatively, you can select the "Outputs" tab and specify the outputs for trimming. In this case, you would usually specify one or more outputs as "Known" and enter their known value in the "Value" column.

Operating Points – Optimization-based Search

- During trimming, a new window shows the current progress (“Trim progress viewer”).



```
Trim progress viewer

Optimizing to solve for all desired  $dx/dt=0$ ,  $x(k+1)-x(k)=0$ , and  $y=y_{des}$ .

(Maximum Error) Block
-----
(5.00000e-03) magball/Magnetic Ball Plant/Current
(1.74525e-10) magball/Magnetic Ball Plant/dhdt
(1.74525e-10) magball/Magnetic Ball Plant/dhdt

Operating point specifications were successfully met.

An operating point op\_trim1 has been created.
```

Operating Points – Optimization-based Search

- After successful trimming, a new operating point is added to the Linear Analysis Tool workspace (not the MATLAB workspace).
- Double-click the new operating point to see its details.

Linear Analysis Workspace	
Name ^	Value
op_trim1	1x1 OperatingReport



Edit: op_trim1				
Optimizer Output		Details		
State	Input	Output		
State	Desired Value	Actual Value	Desired dx	Actual dx
magball/Controller/PID Controller/Filter/Cont. Filter/Filter				
State - 1	[-Inf , Inf]	0	0	0
magball/Controller/PID Controller/Integrator/Continuous/Integrator				
State - 1	[-Inf , Inf]	14.0071	0	0
magball/Magnetic Ball Plant/Current				
State - 1	[0 , Inf]	7.0036	0	4.2064e-11
magball/Magnetic Ball Plant/dhdt				
State - 1	[-Inf , Inf]	0	0	-1.7453e-10
magball/Magnetic Ball Plant/height				
State - 1	0.05	0.05	0	0

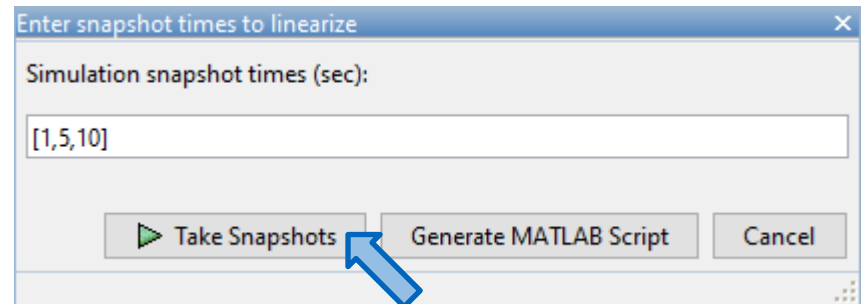
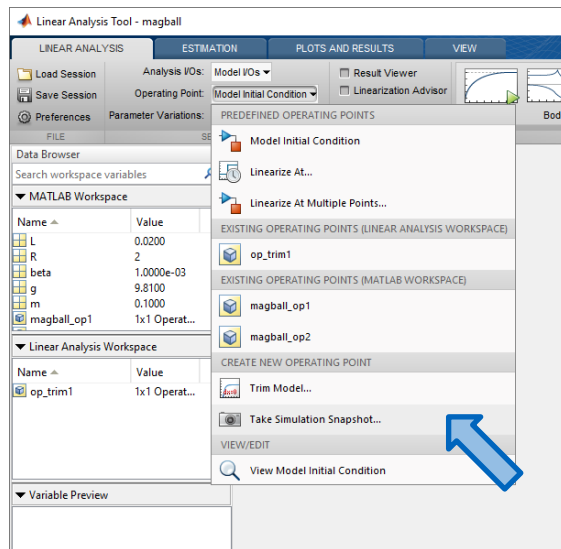
Two state derivatives are not exactly zero, but sufficiently small.

A minimum value of zero had been specified for this state.

This state had been specified as “known.”

Operating Points – Simulation-based Search



- To initialize a steady-state operating point search using simulation snapshot, select “Take Simulation Snapshot...” from the “Operating Point” dropdown menu.



- A new window opens that prompts you to choose one or more simulation snapshot times.
- Click “Take Snapshots” to take snapshots at the specified times.

Operating Points – Simulation-based Search

- The snapshots will then appear in the Linear Analysis Tool workspace. They contain all state values at those times.

▼ Linear Analysis Workspace	
Name ▲	Value
 op_snapshot1	3x1 OperatingPoint
 op_trim1	1x1 OperatingReport



Edit: op_snapshot1

Select Operating Point: Operating point originally taken at t = 5

States Inputs

State	Value
magball/Controller/PID Controller/Filter/Cont. Filter/Filter	
State - 1	3.0138e-07
...ll/Controller/PID Controller/Integrator/Continuous/Integrator	
State - 1	14.0071
magball/Magnetic Ball Plant/Current	
State - 1	7.0036
magball/Magnetic Ball Plant/dhdt	
State - 1	2.5282e-08
magball/Magnetic Ball Plant/height	
State - 1	0.05

Refresh Initialize model...

- The generated snapshots can be used to initialize the optimization-based search (see next slide). However, for this purpose, generate only a single snapshot at a specific time (instead of a vector as shown in the last slide).

Operating Points – Initialize Optimization-based Search with Snapshots

- To initialize the optimization-based search with a snapshot, select “Trim Model...” from the “Operating Points” drop down menu as before.
- However, this time, click “Import...”

Trim the model

SpecificationsOptions

StatesInputsOutputs

State	Value	State Specifications					
		<input type="checkbox"/> Known	<input checked="" type="checkbox"/> Steady State	Minimum	Maximum	dx Minimum	dx Maximum
magball/Controller/PID Controller/Filter/Cont. Filter/Filter							
State - 1	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	-Inf	Inf	-Inf	Inf
magball/Controller/PID Controller/Integrator/Continuous/Integrator							
State - 1	14.0071	<input type="checkbox"/>	<input checked="" type="checkbox"/>	-Inf	Inf	-Inf	Inf
magball/Magnetic Ball Plant/Current							
State - 1	7.0036	<input type="checkbox"/>	<input checked="" type="checkbox"/>	0	Inf	-Inf	Inf
magball/Magnetic Ball Plant/dhdt							
State - 1	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	-Inf	Inf	-Inf	Inf
magball/Magnetic Ball Plant/height							
State - 1	0.05	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	-Inf	Inf	-Inf	Inf

Sync with Model

Import...

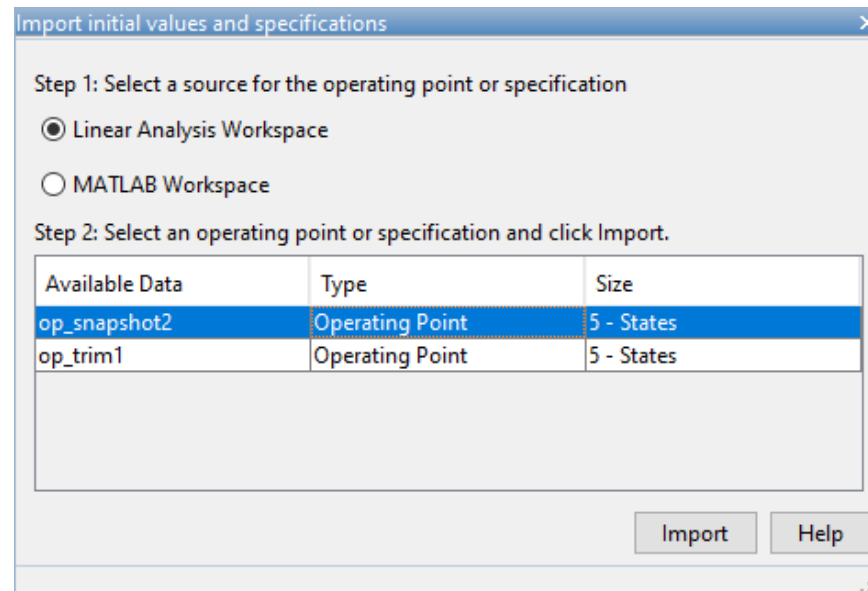
Export...

Start trimming

Generate MATLAB Script

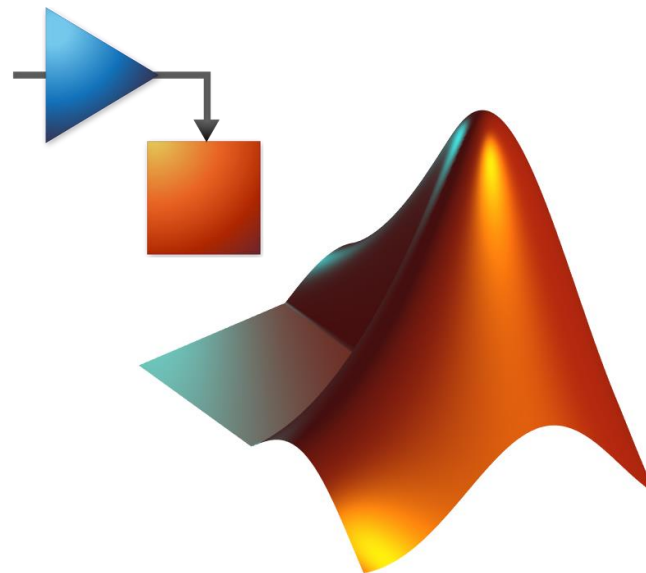
Operating Points – Initialize Optimization-based Search with Snapshots

- Select the snapshot you have just created and click “Import.”



- The state values in the “Trim the model...” window are now the values from the snapshot.
- You can now continue with the optimization-based trimming as described above.
- Instead of using the GUI, you can also write MATLAB code that performs all those same steps automatically. MATLAB code can also be generated from the GUI. Refer to the MATLAB help (function `findop`, Linear Analysis Tool) for more information.

3. Linearization



Linearization

- Simulink itself provides the function `linmod` for model linearization. Again, the Simulink Control Design toolbox brings several advantages, including GUIs and more flexibility.
- It is possible to either compute an exact linearization, or to make a frequency response estimation. Exact linearization is covered in this section. Refer to section 4 for frequency response estimation.

Exact linearization

- Is faster, because it does not require simulation of the Simulink model.
- Returns a parametric (state-space).
- Should be used in most cases.

Frequency Response Estimation

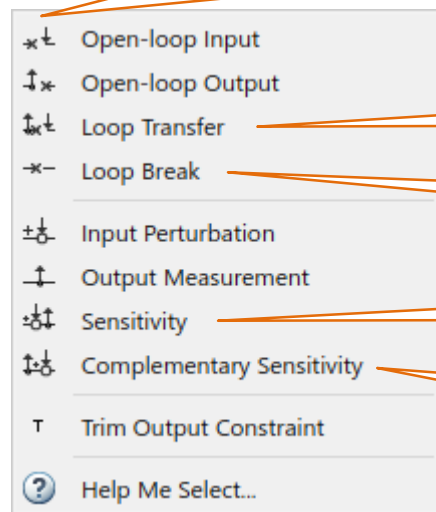
Use this approach

- to validate exact linearization accuracy.
- when the Simulink model contains discontinuities or non-periodic event-based dynamics.
- to study the impact of amplitude size on frequency response.

Linearization

- Before actual linearization, the portion of the model to be linearized has to be specified. This can be done in several ways:
 - In the model, by adding inports, outports, and/or by specifying additional linear analysis points. Note that these points are saved with the model! *Right click on a signal → Linear Analysis Points*

These symbols illustrate how the selected signal will be treated during linearization. They are combinations of loop break, input, output and additive disturbance.



Use to compute open-loop transfer function around the loop.

Use in case of nested loops to ignore the effect of certain loops.

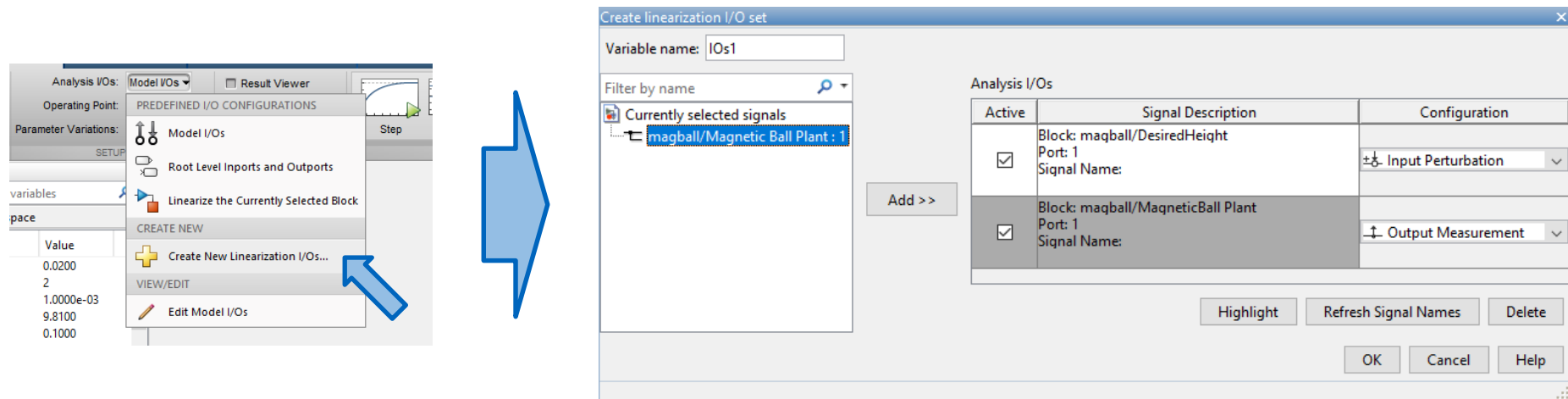
Use to compute sensitivity function for additive disturbance.

Use to compute closed-loop transfer function around the loop.

- Using the Linear Analysis Tool GUI (details on the next slide)
- Using the MATLAB command `linio` (refer to MATLAB help)

Linearization

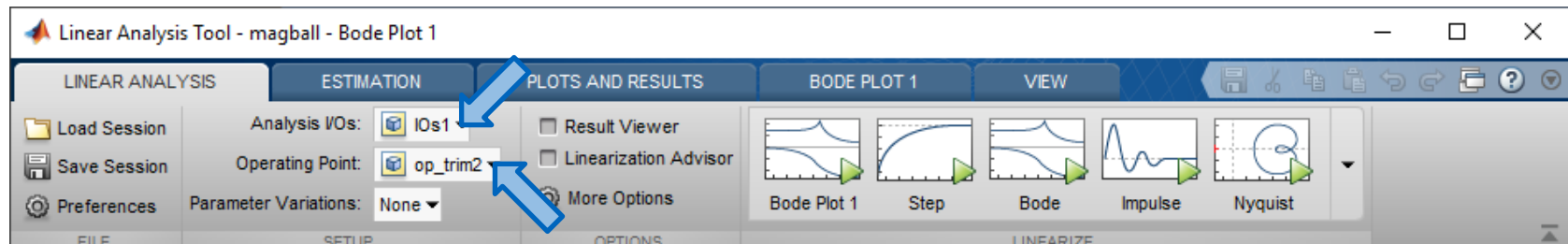
- To specify linear analysis points in the GUI, select “Create New Linearization I/Os...” from the “Analysis I/Os” dropdown menu. The “Create linearization I/O set” window opens.



1. Now, select a signal in the Simulink model (simple left click)
2. It appears in the window. Click “Add >>” to add the signal.
3. Repeat steps 1 and 2 to add further signals, if needed.
4. Use the dropdown menu for specification (same options as above)
5. Click OK to save the I/O set.

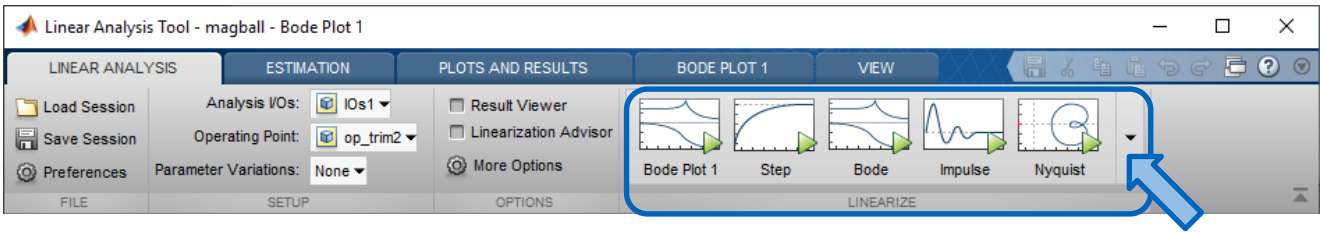
Linearization

- Having defined the operating point as well as the input/output set to linearize, actual linearization can be accomplished with the following steps.
1. In the GUI, select the linearization inputs/outputs set. You can choose, for example, the set you have previously created, the root level in- and outports, or a selected block.
 2. Select an operating point or simulation snapshot. Again, you can choose between predefined operating points, such as the model initial condition, and those operating points you have previously created.



Linearization

- 3. With the proper I/O set and the operating point selected, click on one of the plot symbols to linearize.



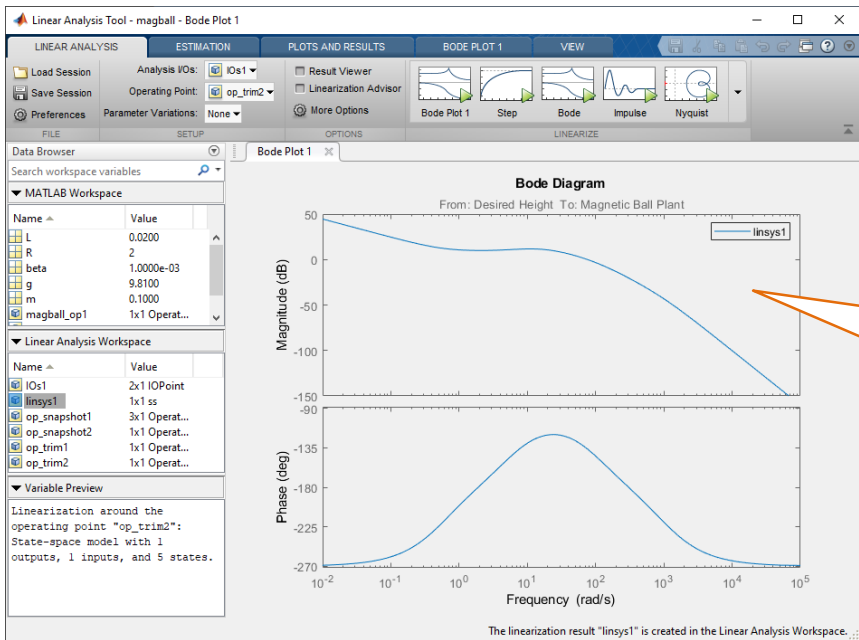
- The linearized model will appear in the GUI workspace and the plot will show on the GUI.

Use “drag and drop” to copy variables from one workspace to another.

The I/O set

The linear system

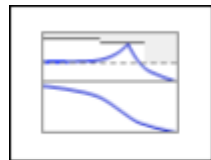
The operating point



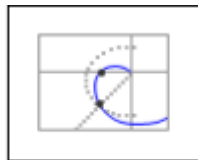
The plot. It can be used for (initial) analysis of the linearized system.

Linearization – Visualization During Simulation

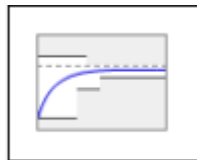
- Linear system characteristics can also be visualized during simulation. They are computed at specified snapshot times, or upon activation by an external trigger.
- In the Simulink Library Browser, under *Simulink Control Design/Linear Analysis Plots*, you can find the following blocks:



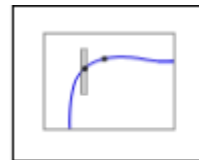
Bode Plot



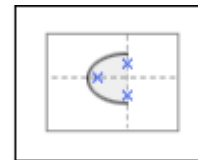
Gain and Phase Margin Plot



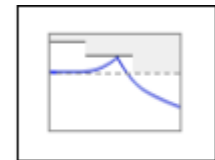
Linear Step Response Plot



Nichols Plot



Pole-Zero Plot



Singular Value Plot

- Place one or more of these blocks in the model to be analyzed.
- To configure a block, open it with a double-click.
- You will see four tabs: “Linearizations”, “Bounds”, “Logging” and “Assertion”

Linearization – Visualization During Simulation

- On the “Linearizations” tab, you can specify the input/output set. To do this, click the “+” symbol to open an additional window. Then proceed like with the Linear Analysis Tool. (Select a signal in the Simulink model, click “<<” and specify configuration in drop down menu.)
- You can also choose whether to linearize on snapshots (and specify the snapshot times), or on an external trigger (the block will then have an inport).
- On the “Logging” tab, you can choose to save the results to the MATLAB workspace.
- An example for the “Bounds” tab:

☒ Include gain and phase margins in assertion

Gain margin (dB) >

Phase margin (deg) >

Feedback sign:

Block Parameters: Nichols Plot

Nichols Plot

Compute and display a linear system on a Nichols plot. You can also specify bounds on the linear system and assert that the bounds are satisfied.

Linearizations Bounds Logging Assertion

Linearization inputs/outputs:

Block : Port : Bus Element	Configuration
scdaircraft/Velocity Controller : 1	Loop Transfer

+

×

↓

⚙

Linearize on:

Snapshot times:

Trigger type:

▶ Algorithm Options

▶ Labels

Show Plot ☐ Show plot on block open ☐ Response Optimization...

?

OK

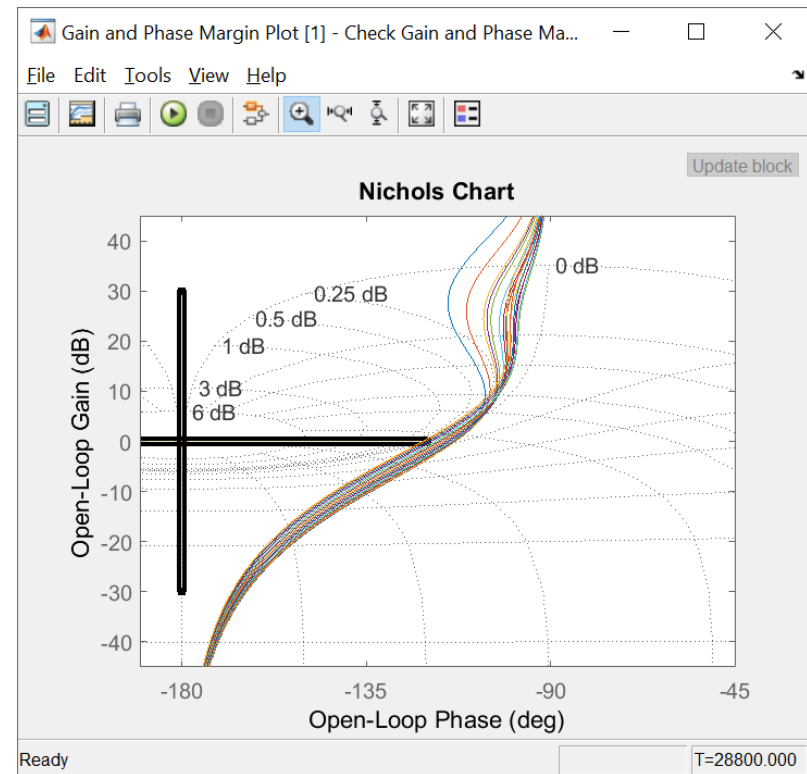
Cancel

Help

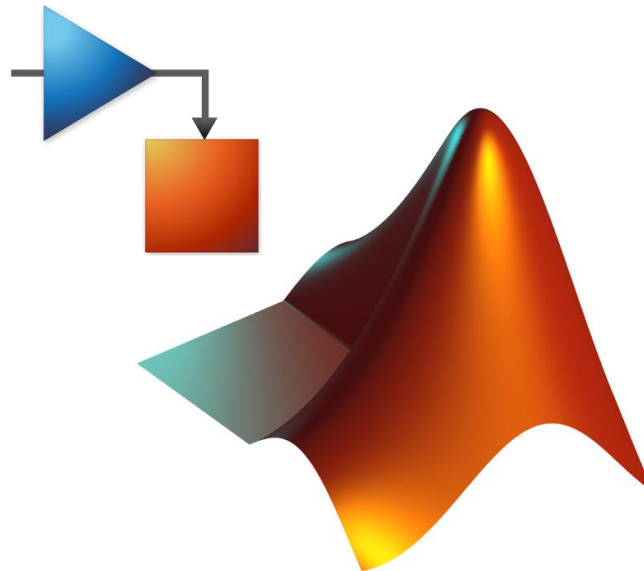
Apply

Linearization – Visualization During Simulation

- On the tab “Bounds,” you can define bounds that are visualized on the plots and that can be used for assertion.
It is often useful to define bounds from design requirements.
- On the tab “Assertion,” you can enable assertion of the bounds. You can also choose
 - to output the assertion signal,
 - to stop the simulation when assertion fails or
 - to make a custom callback when assertion fails.
- To view the plot, click “Show Plot.” Show Plot
- On the right, you can see an example of a gain and phase margin plot, with linearizations at different simulation snapshot times.



4. Frequency Response Estimation



Frequency Response Estimation

- As indicated above, frequency response estimation can be seen as an alternative to exact linearization. It is especially useful
 - to validate exact linearization accuracy,
 - when the Simulink model contains discontinuities or non-periodic event-based dynamics and
 - to study the impact of amplitude size on frequency response.
- The frequency response of a stable system describes the amplitude change and phase shift as a function of frequency (think of the Bode diagram...).
- Just like exact linearization, frequency response estimation requires an input/output set.
- An estimation signal (chirp, sinestream or random) is then injected at the specified input.
- The resulting signal at the specified output is observed
- The estimation is then $G(s) \approx \frac{\text{fast Fourier transform of output signal}}{\text{fast Fourier transform of input signal}}$

Frequency Response Estimation

- Frequency response estimation can be done using the Linear Analysis Tool. Select the “Estimation” tab.
- From the “Input Signal” dropdown menu, select “Sinestream...” The “Create sinestream input” dialog box appears (see next slide).

Signal that is injected for estimation.

Same as for linearization.

Several adjacent sine waves of varying frequency

A sine wave with continuously changing frequency

Image source:
https://de.mathworks.com/help/slcontrol/ug/freestimate_overview.png

Frequency Response Estimation

- How to proceed on the “Create sinestream input” dialog box:

General

Variable name: Frequency units: Simulation order:

☒ Perform filtering to improve estimation results

Parameters

Initialize frequencies and parameters based on a system

System:

There are currently no frequencies specified for the sinestream signal. Add frequencies using add button or initializing them based on a linear system.

10¹ 10² 10³

Frequency (rad/s)

Amplitude:

Number of periods:

Settling periods:

Ramp periods:

Number of samples at each period:

OK

Cancel

Help

Click here to add new frequencies.

Generated frequencies

Add frequencies

Specify by:

From: To:

spaced frequencies

OK

Cancel



Click and drag select frequencies to modify parameter values.

10⁻¹ 10⁰ 10¹ 10²

Frequency (rad/s)

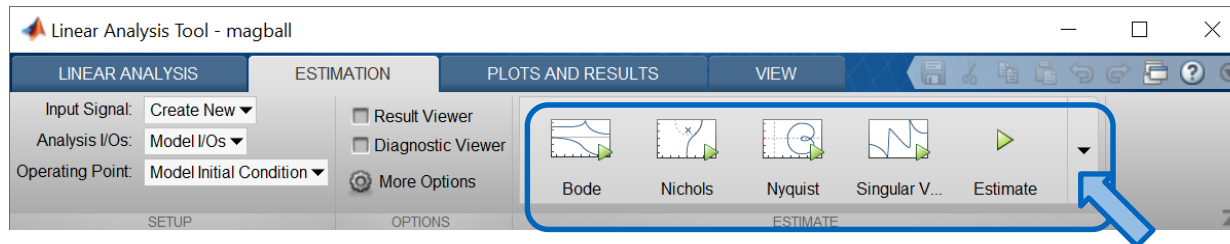
MATLAB / Simulink Lab Course
2021

Lehrstuhl für
Flugsystemdynamik



Frequency Response Estimation

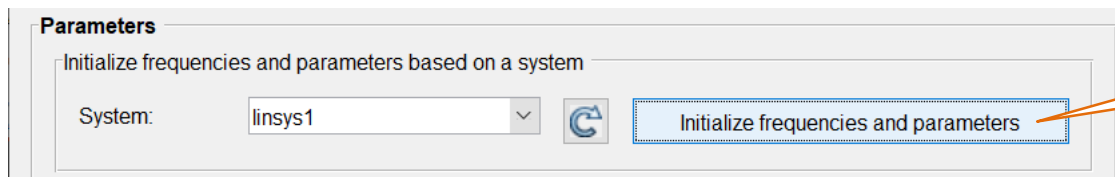
- The next step is to select all frequencies created (the red circles). You can do this like you would select multiple files on your desktop: left-click and hold, draw a rectangle around all red circles and then release the mouse button.
- The boxes below the diagram become active. In the “Amplitude” box, enter 1.
- Click “OK” to create the signal.
- Back on the Linear Analysis Tool, click on one of the plot symbols to start the estimation.



- The estimation model will appear in the Linear Analysis Tool workspace and the plot will show on the GUI.
- Again, you can also use MATLAB code (freqest) or generate MATLAB code from the GUI.

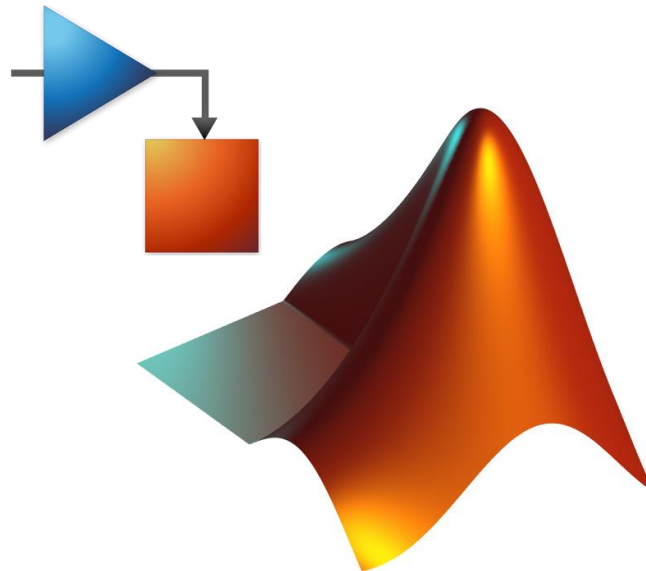
Frequency Response Estimation

- Frequency response estimation is also a useful tool for verification of an exact linearization.
- Suppose you have linearized a system with the Linear Analysis Tool. The resulting linear system is called “linsys1”. There is a Bode plot called “Bode Plot 1”
- You can, again, create a sinestream input. This time, however, you can initialize the frequencies of the signal based on the dynamics of the linear system linsys1.



- To plot the frequency response estimation on top of the existing Bode diagram of linsys1, click on the “Bode Plot 1” button (instead of the “Bode” button).
- Ideally, the frequency response (= Bode plot) of the estimated model matches that of the linearized model linsys1.

5. Control Design



Control Design

- Suppose you have determined operating points of the plant under consideration, linearized the plant and verified the linearization.
- You could now take the linearized state-space model and apply the control design methods of the Simulink Control System toolbox (which you already know).
- You can also use the PID Tuner or the SISO Design Tool from within the Simulink environment (this section).
- Moreover, with the Robust Control Toolbox, you could perform control system tuning using the functions `controlSystemTuner` and `systune` (see MATLAB help).

Control Design – PID Tuner

- To use the PID Tuner from within Simulink for control design, you need to use either the PID Controller or the PID Controller (2DOF) block in your model.
- Open the block dialog (double-click), then click “Tune...” to open the PID Tuner.
- When opening the PID Tuner, the software computes a linearized plant model as seen by the controller (using the current operating point).

Block Parameters: PID Controller

PID 1dof (mask) (link)

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: **PID** Form: **Parallel**

Time domain:

☒ Continuous-time
☐ Discrete-time

Discrete-time settings

Sample time (-1 for inherited): **-1**

Compensator formula

$$P + I \frac{1}{s} + D \frac{N}{1 + N \frac{1}{s}}$$

Main Initialization Output Saturation Data Types State Attributes

Controller parameters

Source: **internal**

Proportional (P): **1**

Integral (I): **1**

Derivative (D): **0**

☒ Use filtered derivative

Filter coefficient (N): **100**

Automated tuning

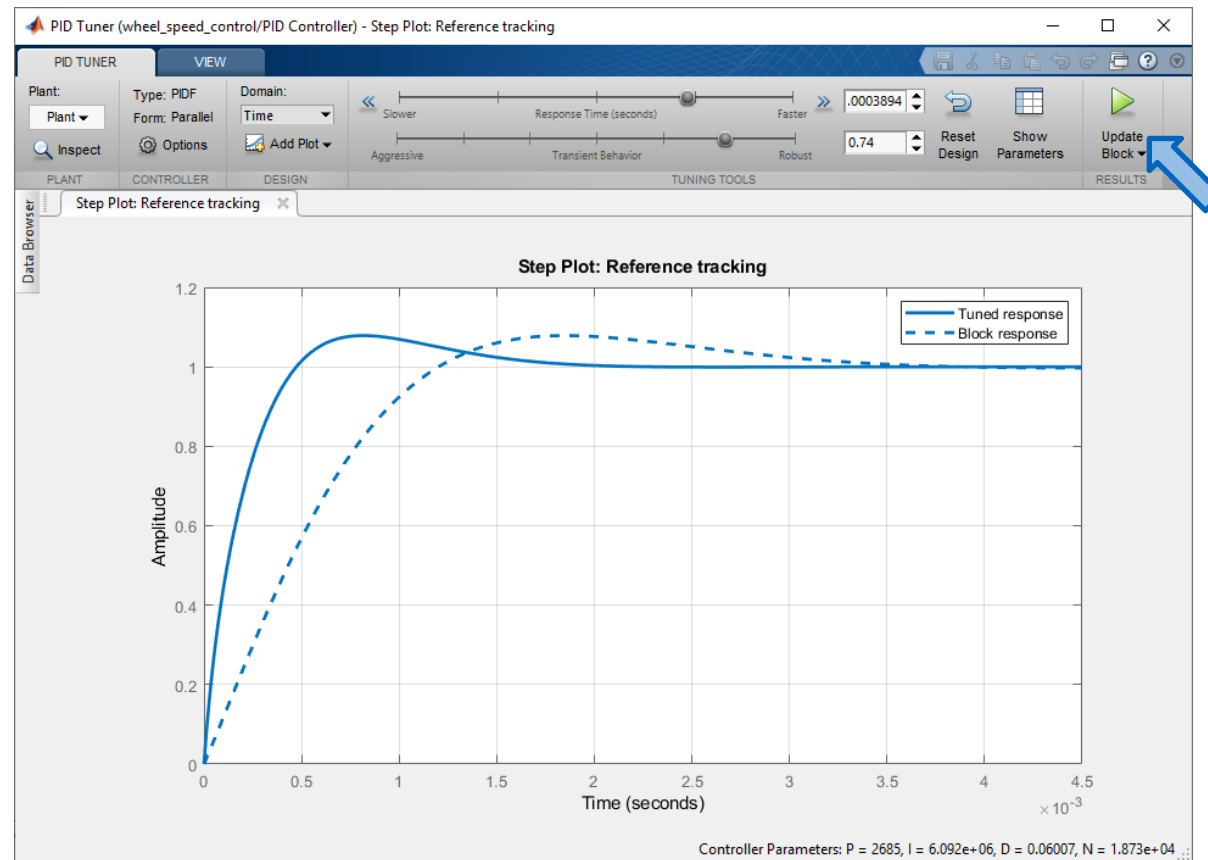
Select tuning method: **Transfer Function Based (PID Tuner App)** **Tune...**

☒ Enable zero-crossing detection

OK Cancel Help Apply

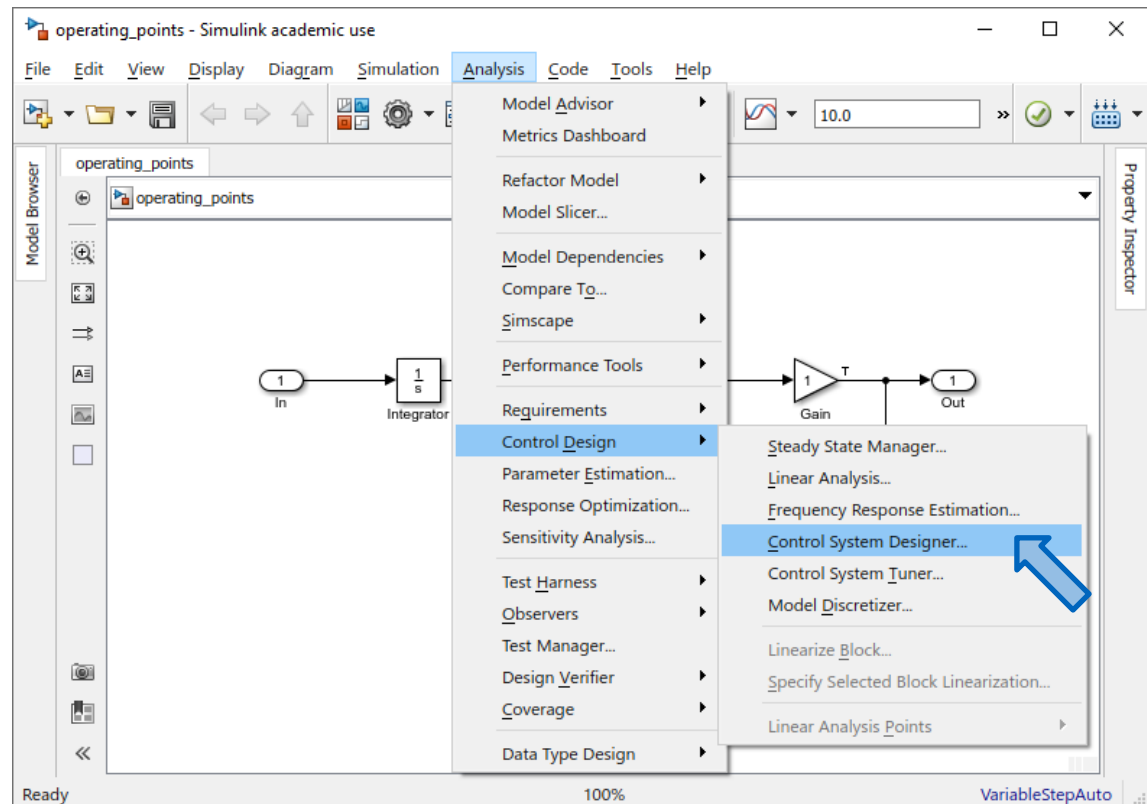
Control Design – PID Tuner

- Once the desired result is obtained, click “Update Block” to automatically adjust the block settings according to your tuning result.



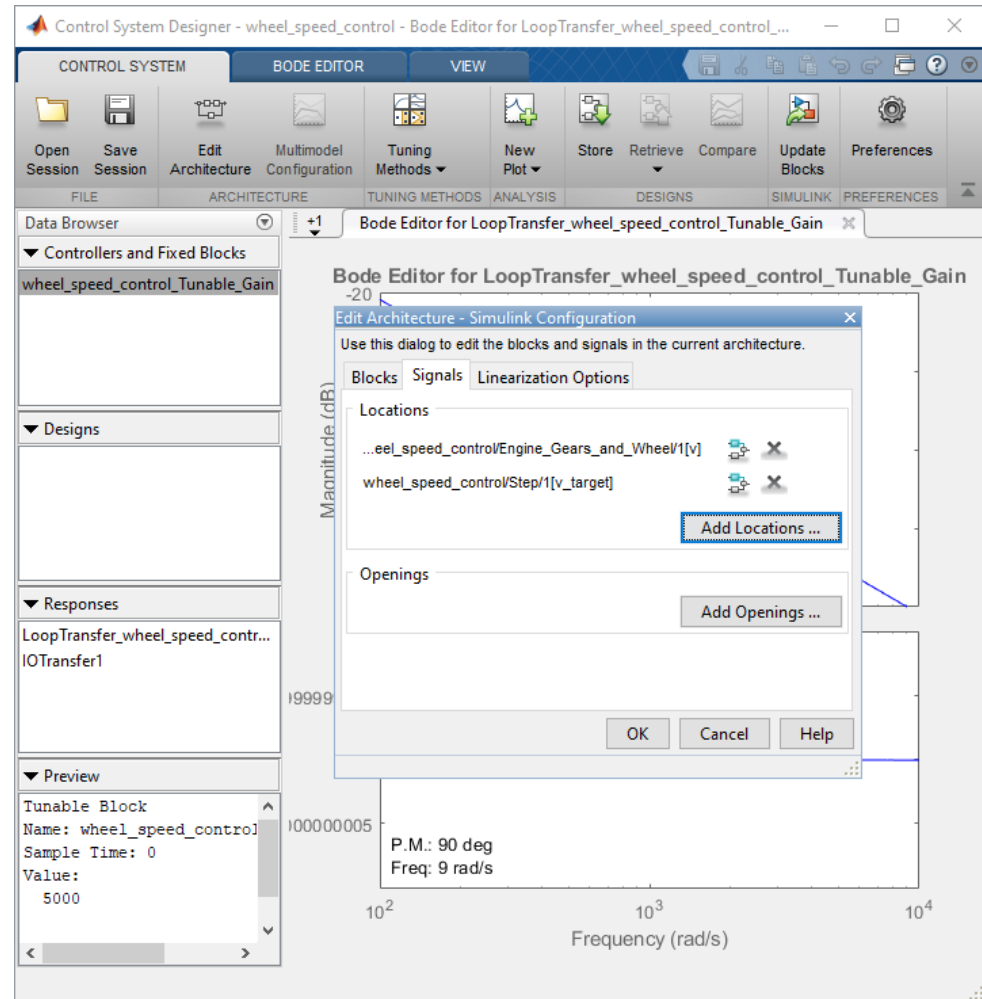
Control Design – SISO Loop Tuning

- The SISO Design Tool as well can be used from within the Simulink environment. Note that it only works with tunable blocks in the model:
 - Gain
 - LTI System
 - Discrete Filter
 - PID Controller
 - State-space blocks
 - Zero-pole blocks
 - Transfer function blocks
- The first step is to open the Control and Estimation Tools Manager by proceeding as shown on the right (select “Control System Designer...”).



Control Design – SISO Loop Tuning

- In the Control System Designer, add the tunable blocks under “Edit Architecture” > “Blocks”.
- Under “Signals”, add the Simulink signals that you want to use for tuning your controller (linearization).
- Further options for the linearization of the system, such as operating points, can be configured.
- Various graphical tuning methods and analysis plots are available (see next slide).

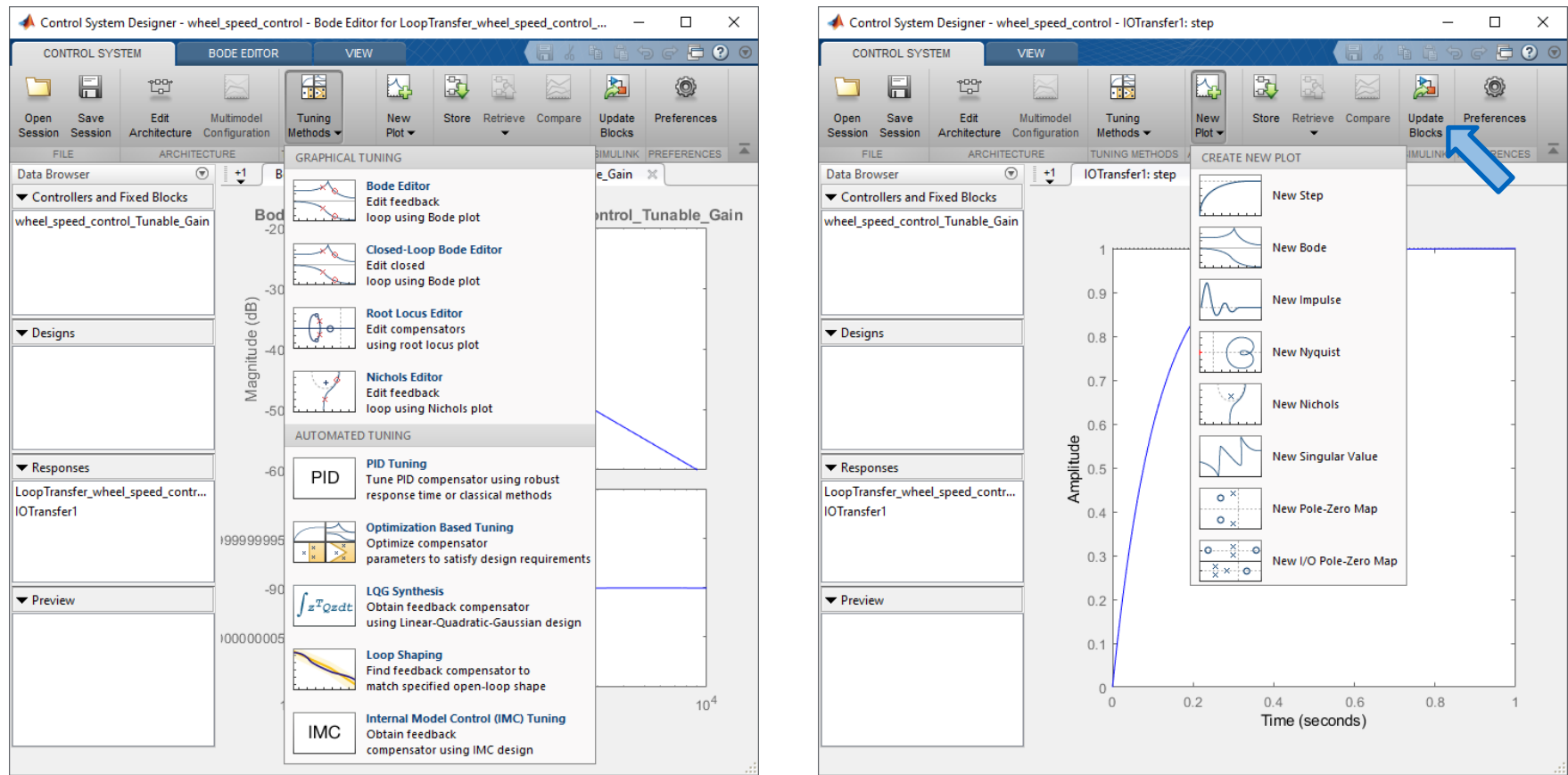


Control Design – SISO Loop Tuning

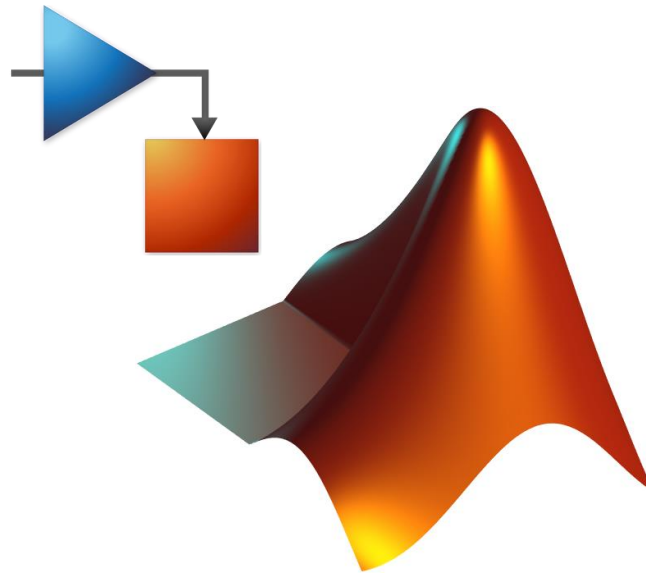
- Graphical **tuning methods** are interactive plots, which are used to tune the response of the system using either open-loop or closed-loop design methods. With the tuning methods, you add, remove and modify controller poles, zeros and gains:
 - Bode Editor, Close-Loop Bode Editor
 - Root Locus Editor
 - Nichols Editor
- Additional **plots** allow to analyze the response or dynamics of a closed-loop system in the model. Analysis plots cannot directly be edited, however, will automatically update to reflect the effects of any changes you make in the tuned parameters:
 - Step, Bode, Impulse, Nyquist, Nichols, Singular Value, Pole-Zero Map, I/O Pole-Zero Map
- Additionally, further **automated tuning methods** are available:
 - PID Tuning
 - Optimization Based Tuning
 - LQG Synthesis
 - Loop Shaping
 - Internal Model Control (IMC) Tuning

Control Design – SISO Loop Tuning

- Once you have obtained the final design, simply click on “Update Simulink Block Parameters” to automatically adjust the block settings according to your design.

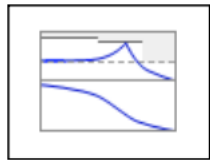


6. Model Verification

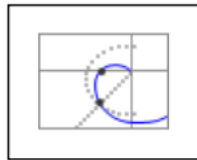


Model Verification

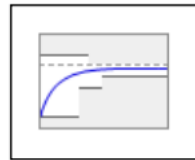
- Model verification can be useful to ensure that a (nonlinear) closed control loop that you have previously designed with linear methods behaves like required.
- Blocks for model verification concerning linear analysis can be found in the Simulink Library Browser, under *Simulink Control Design/Model Verification*.



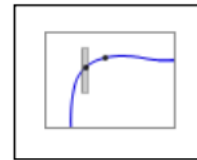
Check Bode Characteristics



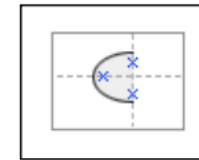
Check Gain and Phase Margins



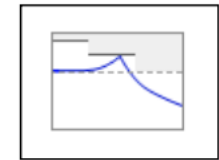
Check Linear Step Response Characteristics



Check Nichols Characteristics



Check Pole-Zero Characteristics

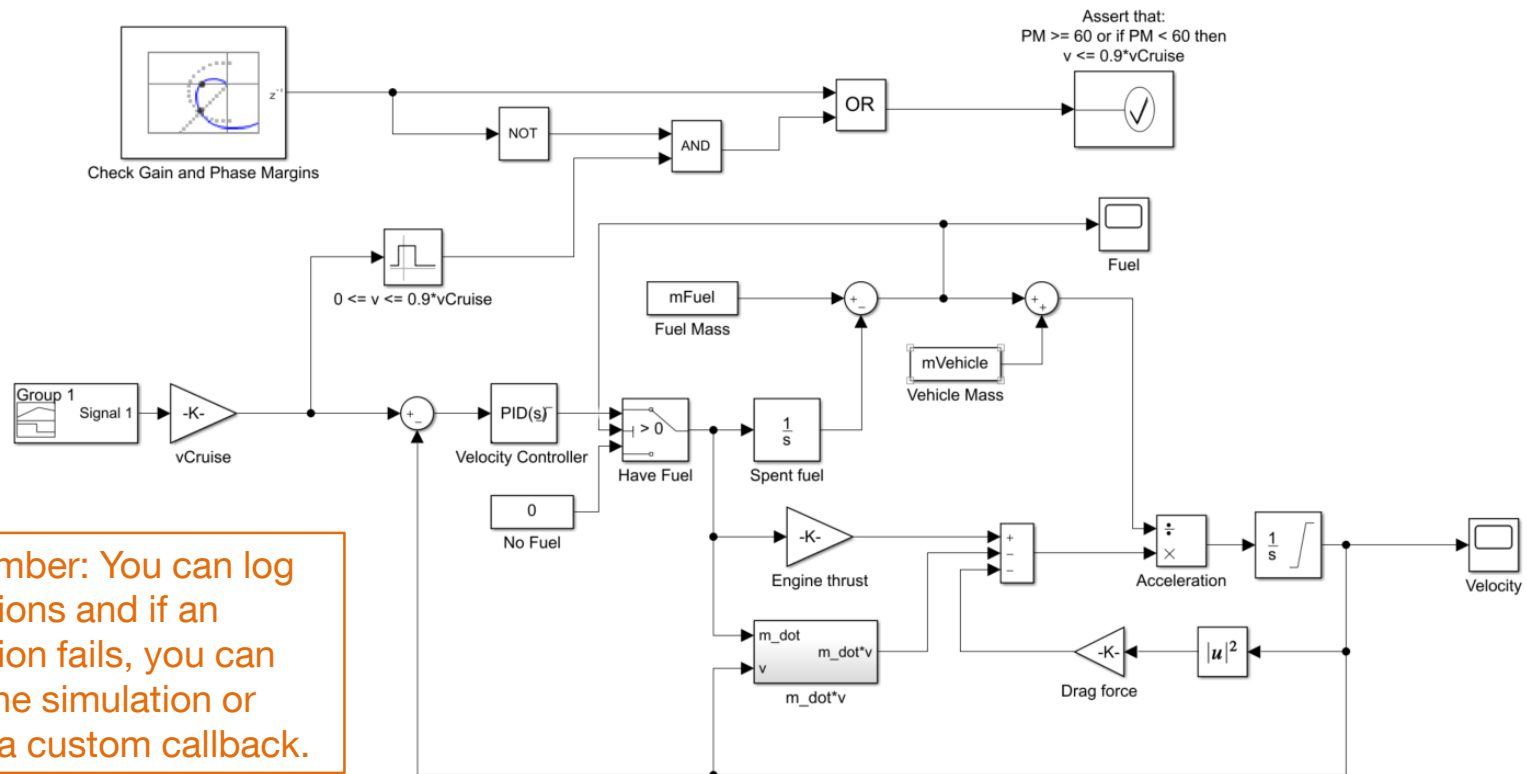


Check Singular Value Characteristics

- These blocks are similar to the Linear Analysis Plot blocks (those for visualization during simulation – see section 3), but have different default settings.
- Most notably, these blocks have default bounds and assertion settings.

Model Verification

- Other blocks for model verification (other than linear analysis) can be found in the Simulink Library Browser under *Simulink/Model Verification*. You can combine these blocks with those from the Simulink Control Design toolbox to model most if not all requirements.



Remember: You can log assertions and if an assertion fails, you can stop the simulation or make a custom callback.

Copyright 2010-2011 The MathWorks, Inc., modified by Institute of Flight System Dynamics, Technical University of Munich, 2019