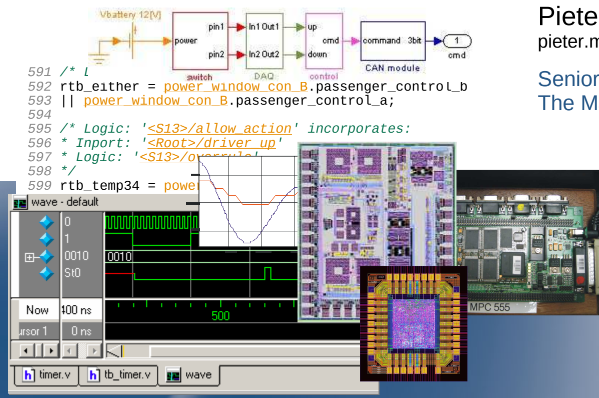
The MathWorks MATLAB & SIMULINK

Model-Based Design—Unpolished



Pieter J. Mosterman
pieter.mosterman@mathworks.com

Senior Research Scientist
The MathWorks, Inc.

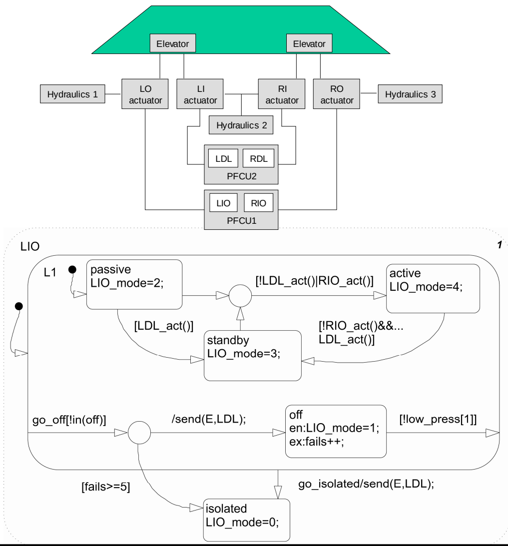
The MathWorks MATLAB & SIMULINK

Agenda

- Requirements engineering
- Feedback control
  - Control law design
  - Evaluation criteria
- Adaptive control
  - Parameter adaptation
  - Switched control
- Hierarchical control
- Design
  - Functional control design flow
  - Implementation
  - Platform based design
- Wrap up

2

Elevator redundancy management



Stateflow (truth table) mode_logic/Mode Logic SS/Mode Logic Chart.L_switch*

File Edit Tools Diagnostics Help

Condition Table:

	Description	Condition	D1	D2	D3	D4	D5	D6	D7
1	Hydraulic system 1 Low pressure (Left Outer line)	low_press[1]	T	T	F	F	-	-	-
2	Left Outer actuator position failed	L_pos_fail[1]	-	-	T	T	-	-	-
3	Hydraulic system 2 Low pressure (Inner line)	low_press[2]	F	-	F	-	T	-	-
4	Left Inner actuator position failed	L_pos_fail[2]	F	-	F	-	-	T	-
		Actions: Specify a row from the Action Table	2	3,5	3	3,5	4	5	Default

Action Table:

#	Description	Action
1	Default - All ok, do nothing.	Default:
2	Hydraulic System 1 Failure. Turn off Left Outer Actuator	send(go_off,Actuators.LIO);
3	Left Outer Actuator Failure. Isolate Left Outer Actuator	send(go_isolated,Actuators.LIO);
4	Hydraulic System 2 Failure. Turn off Left Inner Actuator	send(go_off,Actuators.LDL);
5	Left Inner Actuator Failure. Isolate Left Inner Actuator	send(go_isolated,Actuators.LDL);

Reconfiguration requirements

2.1.11 Hydraulic pressure 1 and Left Outer actuator position failures

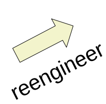
If a failure is detected in both the hydraulic pressure 1 system and the Left Outer actuator position sensor, while there are no other failures, isolate the fault by switching the Left Outer actuator to the off mode.

2.1.12 Hydraulic pressure 2 and Left Inner actuator position failures

If a failure is detected in both the hydraulic pressure 2 system and the Left Inner actuator position sensor, while there are no other failures, isolate the fault by switching the Left Inner actuator to the off mode.

2.1.15 Multiple failures on Left hydraulics and actuators

If multiple failures are detected in hydraulic pressure 1 **or** 2 systems **and either** the Left Outer actuator position sensor **or** the Left Inner actuator position sensor, isolate the fault by switching **both** the Left Outer actuator **and** the Left Inner Actuator to the isolate mode.



2.1.15 Multiple failures on Left hydraulic and actuators

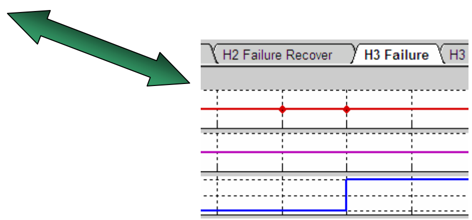
If multiple failures are detected in left hydraulic pressure and actuator positions, isolate the fault by switching both the Left Outer actuator and the Left Inner actuator to the isolated mode. The following combinations trigger this condition:

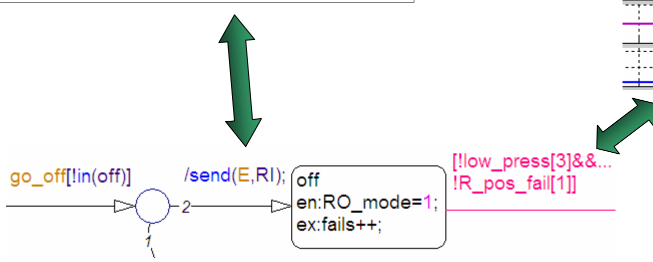
- Failures in hydraulic pressure 1, and hydraulic pressure 2
- Failures in Left Outer actuator position sensor, and Left Inner actuator position sensor
- Failures in hydraulic pressure 1, and Left Inner actuator position sensor
- Failures in hydraulic pressure 2, and Left Outer actuator position sensor

The MathWorks MATLAB&SIMULINK

Requirements traceability

2.1.5 Hydraulic pressure 3 failure
If a failure is detected in the hydraulic pressure 3 system, while there are no other failures, isolate the fault by switching the Right Outer actuator to the off mode.





5

The MathWorks MATLAB&SIMULINK

Agenda

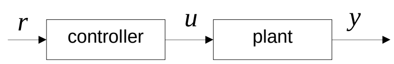
- Requirements engineering
- Feedback control
  - Control law design
  - Evaluation criteria
- Adaptive control
  - Parameter adaptation
  - Switched control
- Hierarchical control
- Design
  - Functional control design flow
  - Implementation
  - Platform based design
- Wrap up

6

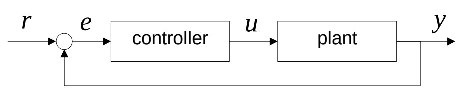
The MathWorks MATLAB&SIMULINK®

Control design architectures

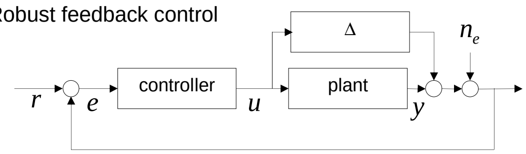
Feedforward control *(experiment in pool room)*

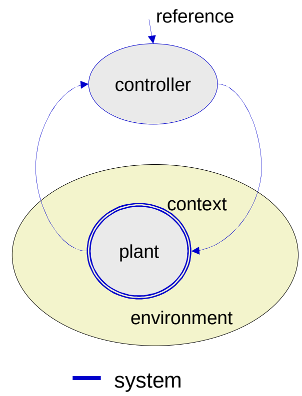


Feedback control *(experiment in wine cellar)*



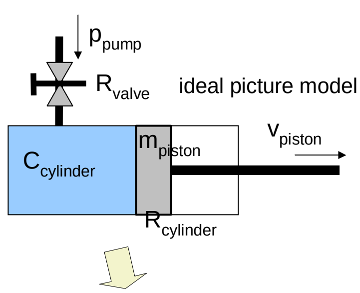
Robust feedback control





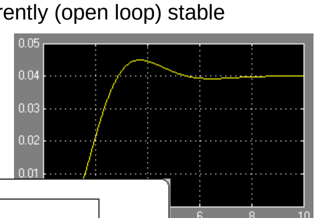
The MathWorks MATLAB&SIMULINK®

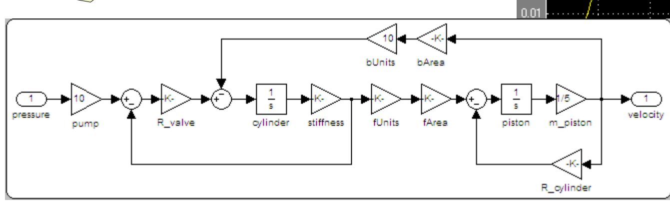
A hydraulic cylinder



ideal picture model

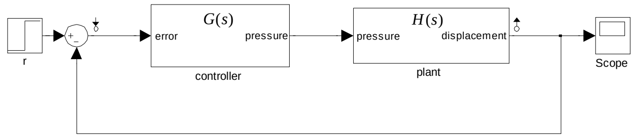
inherently (open loop) stable





second order model of dynamics

Feedback control system



straightforward:
position control

$$G(s) = K_p$$

performance:
position-integral-derivative (PID) control

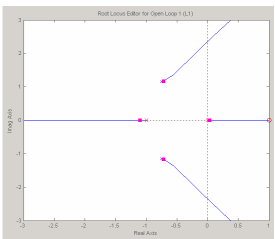
$$G(s) = K_p + \frac{K_i}{s} + sK_d$$

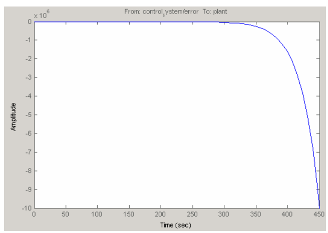
$$= \frac{s^2 K_d + sK_p + K_i}{s}$$

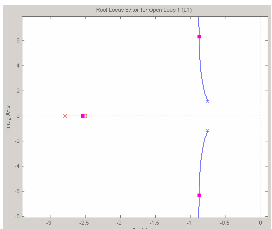
closed loop analysis

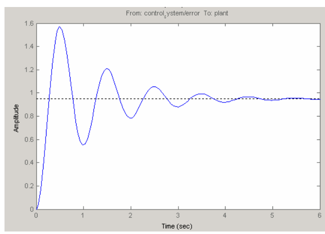
$$Y(s) = \frac{H(s)G(s)}{I + H(s)G(s)} U(s)$$

Compensator pole placement



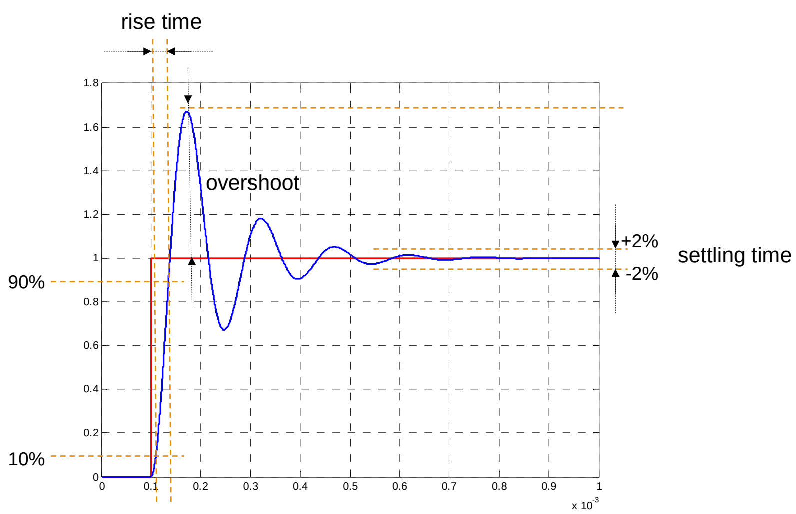






The MathWorks MATLAB&SIMULINK

Performance criteria



11

The MathWorks MATLAB&SIMULINK

Stability classes

Lyapunov stable

$$(\forall \varepsilon > 0)(\exists \delta > 0)(\forall t \geq 0)(\|x(0)\| < \delta \Rightarrow \|x(t)\| < \varepsilon)$$

Asymptotically stable

$$\textit{Lyapunov stable and } (\exists \delta > 0)(\|x(0)\| < \delta \Rightarrow \lim_{t \to \infty} x(t) = 0)$$

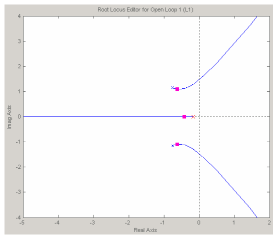
Exponentially stable

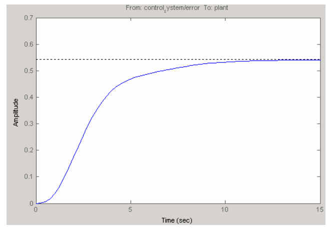
$$(\exists \alpha, \beta, \delta > 0)(\forall t \geq 0)(\|x(0)\| < \delta \Rightarrow \|x(t)\| \leq \alpha \|x(0)\| e^{-\beta t})$$

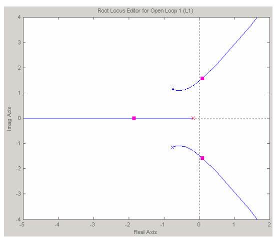
12

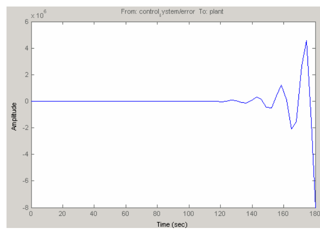
The MathWorks MATLAB&SIMULINK

Robustness analysis









13

The MathWorks MATLAB&SIMULINK

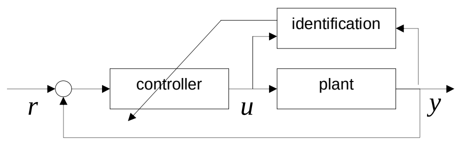
Agenda

- Requirements engineering
- Feedback control
  - Control law design
  - Evaluation criteria
- Adaptive control
  - Parameter adaptation
  - Switched control
- Hierarchical control
- Design
  - Functional control design flow
  - Implementation
  - Platform based design
- Wrap up

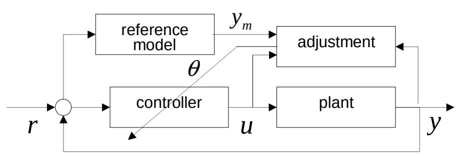
14

Adaptive control

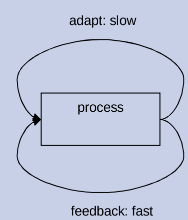
model identification adaptive control



model reference adaptive control



Principle



performance criterion: $J(\theta) = \frac{1}{2} e^2$

adaptation strategy: $\frac{d\theta}{dt} = -\gamma \frac{\partial J}{\partial \theta} = -\gamma \frac{\partial J}{\partial e} \frac{\partial e}{\partial \theta} = -\gamma e \frac{\partial e}{\partial \theta}$

First order system

plant

$$\frac{dy}{dt} = -ay + bu$$

reference model

$$\frac{dy_m}{dt} = -a_m y_m + b_m u_c$$

controller

$$u(t) = \theta_1 u_c(t) - \theta_2 y(t)$$

ideal parameters

$$\theta_1 = \theta_1^0 = \frac{b_m}{b}$$

$$\theta_2 = \theta_2^0 = \frac{a_m - a}{b}$$

error

$$e = y - y_m \qquad y = \frac{b\theta_1}{s + a + b\theta_2} u_c$$

$$\frac{\partial e}{\partial \theta_1} = \frac{b}{s + a + b\theta_2} u_c$$

$$\frac{\partial e}{\partial \theta_2} = -\frac{b^2 \theta_1}{(s + a + b\theta_2)^2} u_c = -\frac{b}{s + a + b\theta_2} y$$

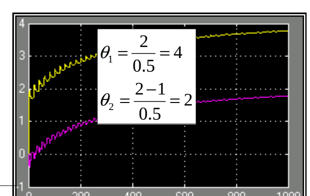
approximate

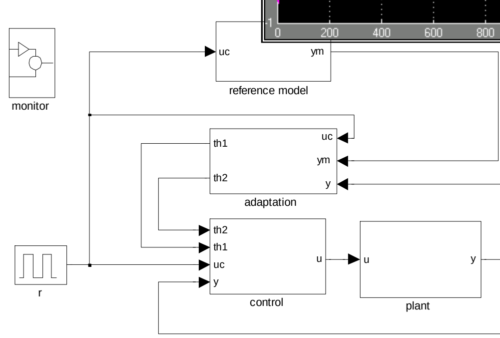
$$s + a + b\theta_2 \approx s + a_m$$

adaptation

$$\frac{d\theta_1}{dt} = -\gamma \left(\frac{a_m}{s + a_m} u_c \right) e$$

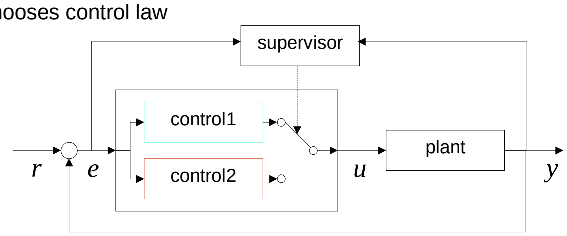
$$\frac{d\theta_2}{dt} = \gamma \left(\frac{a_m}{s + a_m} y \right) e$$



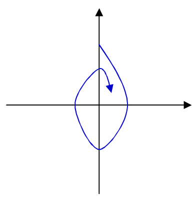


Switched control

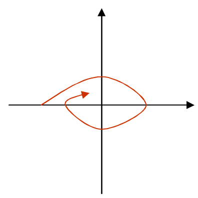
supervisor chooses control law



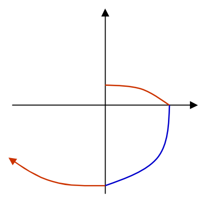
the act of switching may cause unstable compound behavior



control1: stable



control2: stable



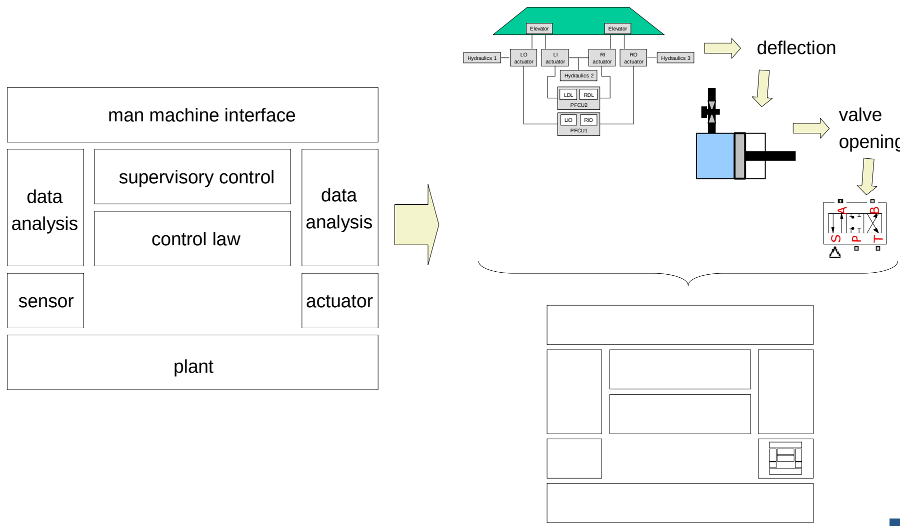
switched control: unstable

Agenda

- Requirements engineering
- Feedback control
  - Control law design
  - Evaluation criteria
- Adaptive control
  - Parameter adaptation
  - Switched control
- Hierarchical control
- Design
  - Functional control design flow
  - Implementation
  - Platform based design
- Wrap up

The MathWorks MATLAB&SIMULINK

Hierarchy in control



19

The MathWorks MATLAB&SIMULINK

Agenda

- Requirements engineering
- Feedback control
  - Control law design
  - Evaluation criteria
- Adaptive control
  - Parameter adaptation
  - Switched control
- Hierarchical control
- Design
  - Functional control design flow
  - Implementation
  - Platform based design
- Wrap up

20

The MathWorks — MATLAB&SIMULINK

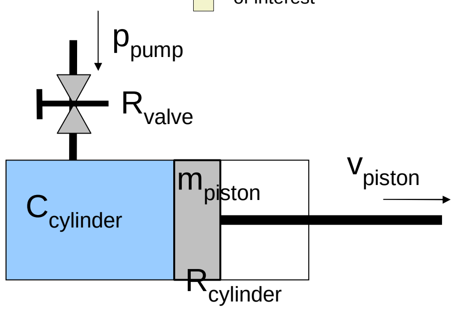
Control synthesis

$$\dot{x}(t) = f(x,u,t)$$
$$0 = g(x,u,t)$$
$$y(t) = h(x,u,t)$$

derive representation amenable to synthesis

$$\begin{cases}\dot{x}_0(t) = A_0 x_0(t) + B_0 u_0(t)\\ y_0(t) = C_0 x_0(t) + D_0 u_0(t)\end{cases}$$
$$\vdots$$
$$\begin{cases}\dot{x}_n(t) = A_n x_n(t) + B_n u_n(t)\\ y_n(t) = C_n x_n(t) + D_n u_n(t)\end{cases}$$

model phenomena of interest



design control law

$$u_i(t) = K_i x_i(t)$$

validate control

$$\begin{cases}\dot{x}_i(t) = (A_i - B_i K_i) x_i(t)\\ y_i(t) = (C_i - D_i K_i) x_i(t)\end{cases}$$

21

The MathWorks — MATLAB&SIMULINK

Functional control design stages

validate

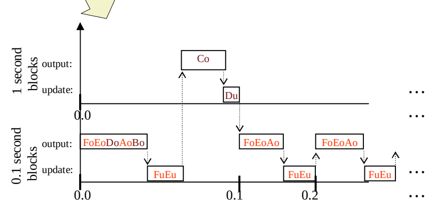
verify

$$\begin{cases}\dot{x}(t) = Ax(t) + Bu(t)\\ y(t) = Cx(t) + Du(t)\end{cases}$$

discretize

$$\begin{cases}x(t_{k+1}) = \tilde{A}x(t_k) + \tilde{B}u(t_k)\\ y(t_k) = \tilde{C}x(t_k) + \tilde{D}u(t_k)\end{cases} \quad \begin{bmatrix}(I+\frac{T}{2}A)(I-\frac{T}{2}A)^{-1} & \frac{4k}{T\sqrt{T}}B(I-\frac{T}{2}A)^{-1}\\ \sqrt{T}C(I-\frac{T}{2}A)^{-1} & \frac{T}{2}D(I-\frac{T}{2}A)^{-1}\end{bmatrix}$$

task assignment



2.1.15 Multiple failures on Left hydraulics and actuators

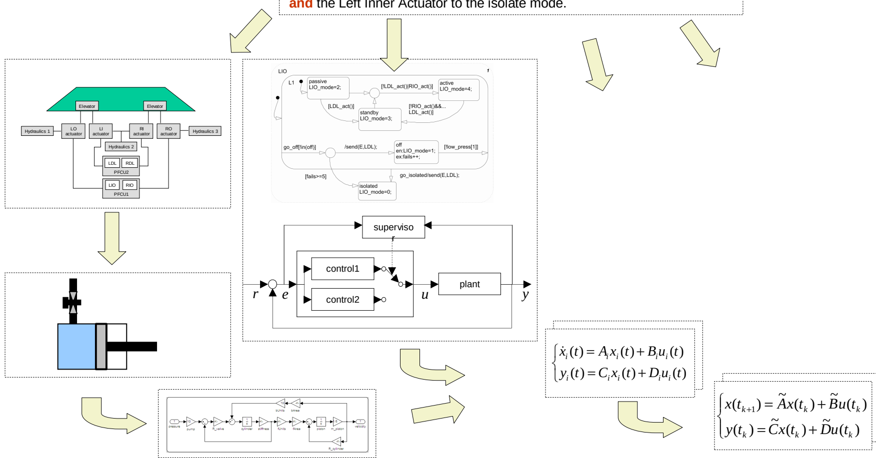
If multiple failures are detected in hydraulic pressure 1 **or** 2 systems **and either** the Left Outer actuator position sensor **or** the Left Inner actuator position sensor, isolate the fault by switching **both** the Left Outer actuator **and** the Left Inner Actuator to the isolate mode.

22

Specifications become requirements

2.1.15 Multiple failures on Left hydraulics and actuators

If multiple failures are detected in hydraulic pressure 1 **or** 2 systems **and** **either** the Left Outer actuator position sensor **or** the Left Inner actuator position sensor, isolate the fault by switching **both** the Left Outer actuator **and** the Left Inner Actuator to the isolate mode.



Agenda

- Requirements engineering
- Feedback control
  - Control law design
  - Evaluation criteria
- Adaptive control
  - Parameter adaptation
  - Switched control
- Hierarchical control
- Design
  - Functional control design flow
  - Implementation
  - Platform based design
- Wrap up

What can we learn?

- Standard input performance excitation
  - Step input
- Standard architectures and terminology
- Standard design representations
  - Root locus
  - Bode plots
  - ➢ Practitioners can relate to theory (relatively) easy
- Well-defined analyses
  - Stability
  - Robustness
- Focused design stages
  - Strict separation
- …

Wrap up

- Offline design is very involved
  - Some aspects highlit
- Adaptation does some design online
  - Which aspects of design can be automated?

- To be continued …