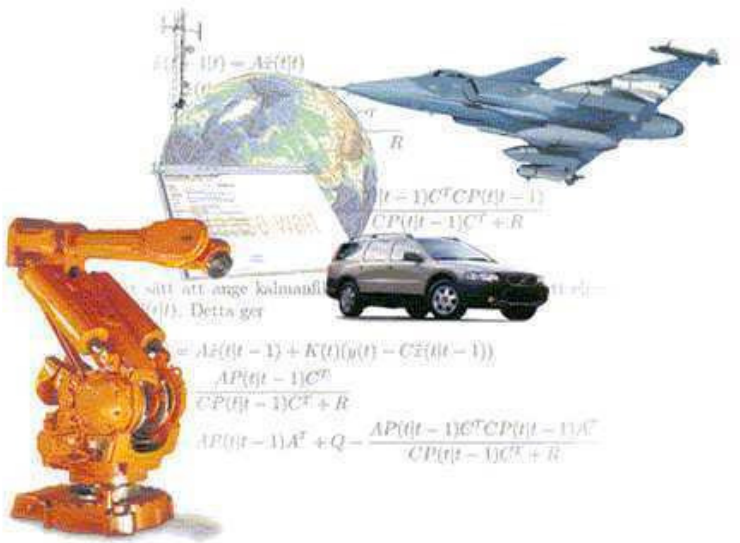
System Identification for Control and Simulation.



Lennart Ljung

Automatic Control, ISY, Linköpings Universitet

Outline

Two goals

- A general overview of basic steps in system identification

Outline

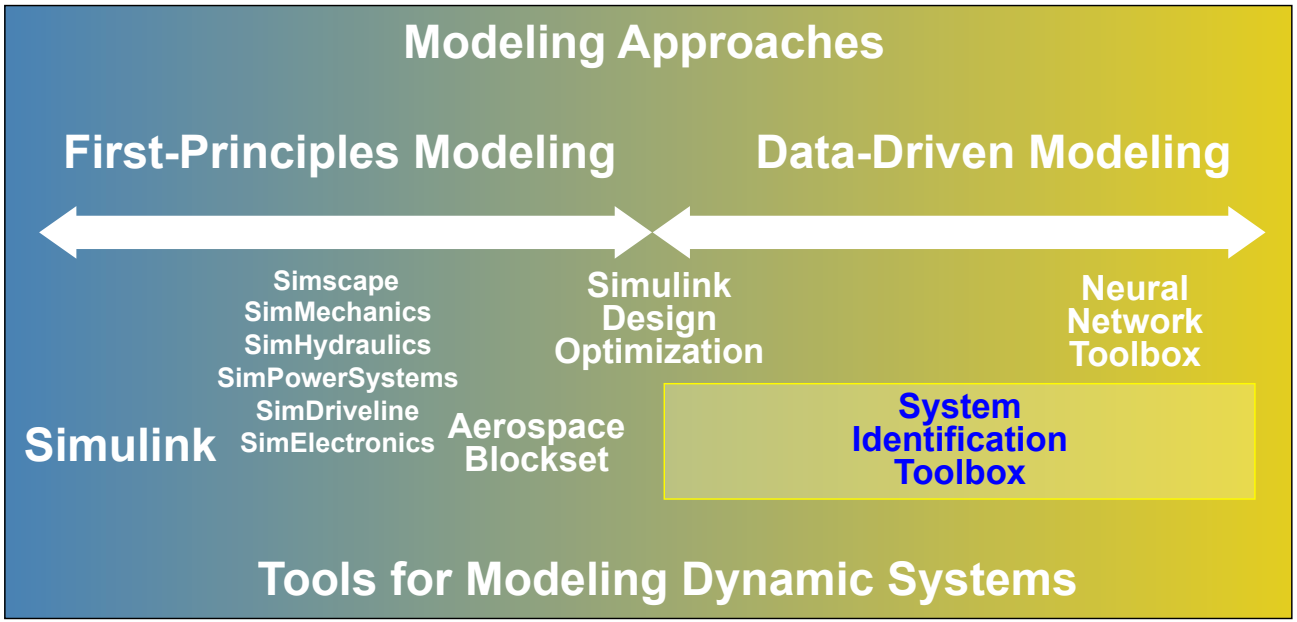
Two goals

- A general overview of basic steps in system identification
- A more technical account of modeling goals for linear systems

Modeling Approaches: View from the Mathworks

MathWorks MATLAB&SIMULINK

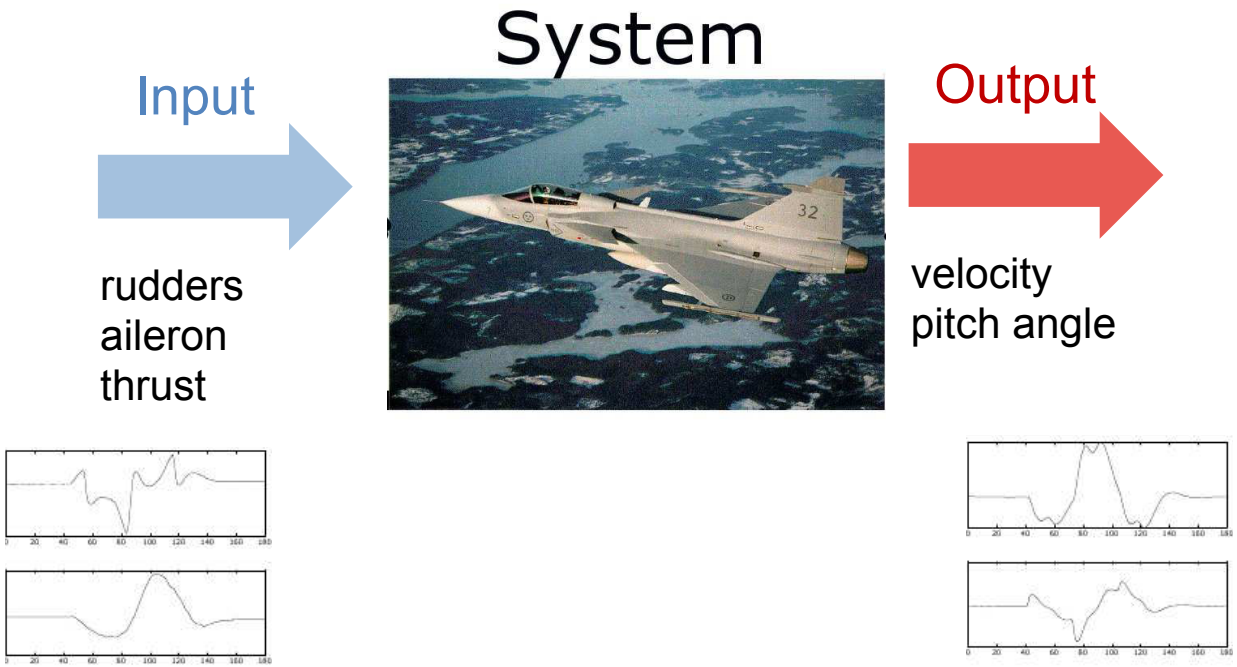
Modeling Dynamic Systems



2

An Introductory Example: System

The System

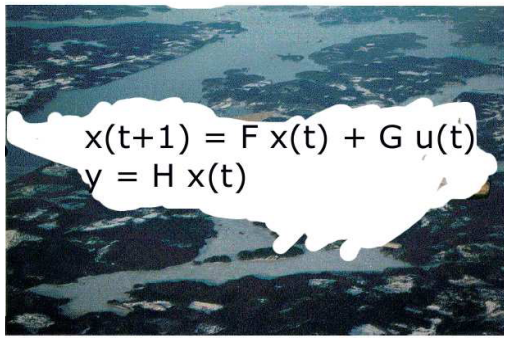


An Introductory Example 2: Model

The Model

u
Input

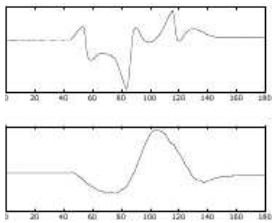
MODEL



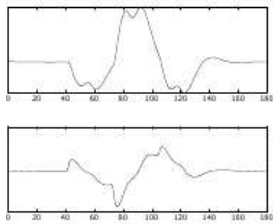
y
Output

rudders
aileron
thrust

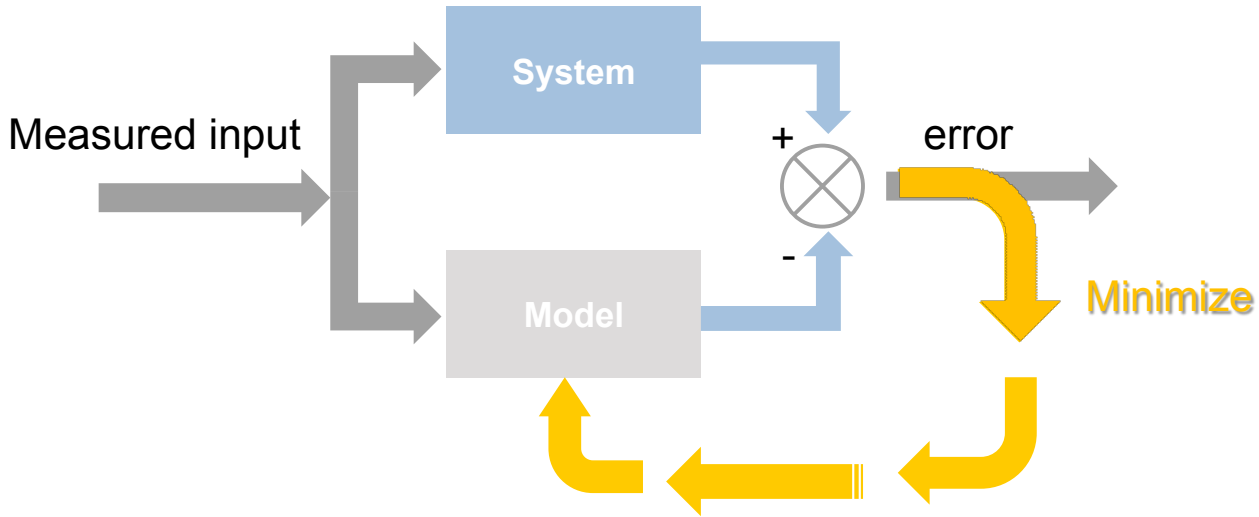
velocity
pitch angle



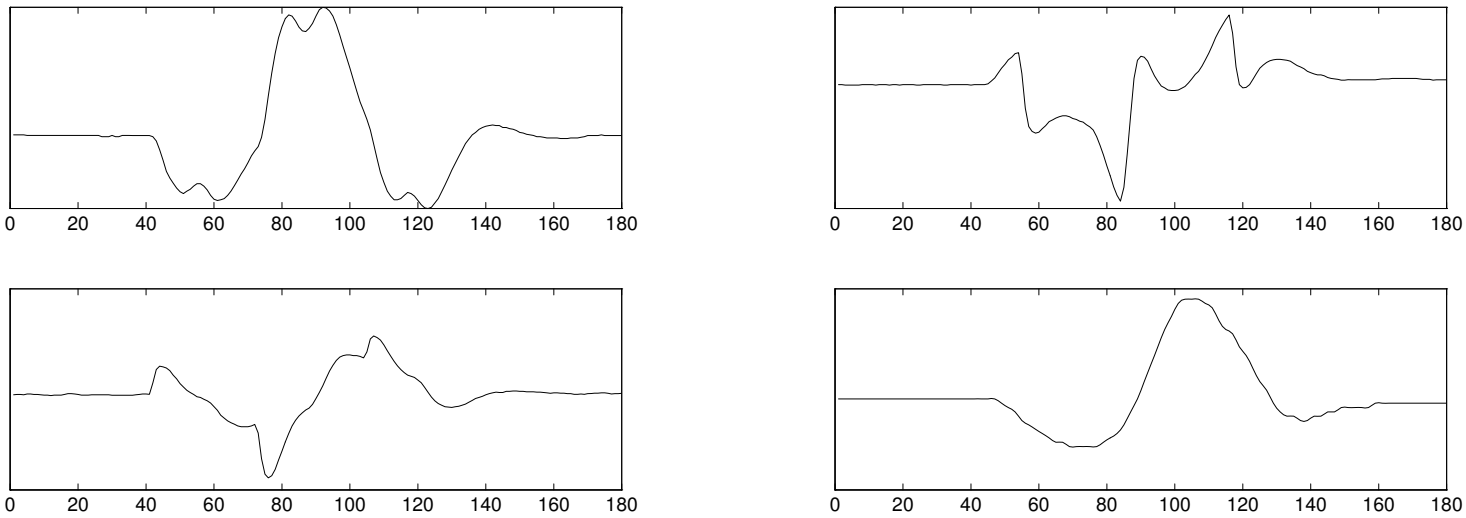
u, y: measured time or frequency domain signals



The System and the Model



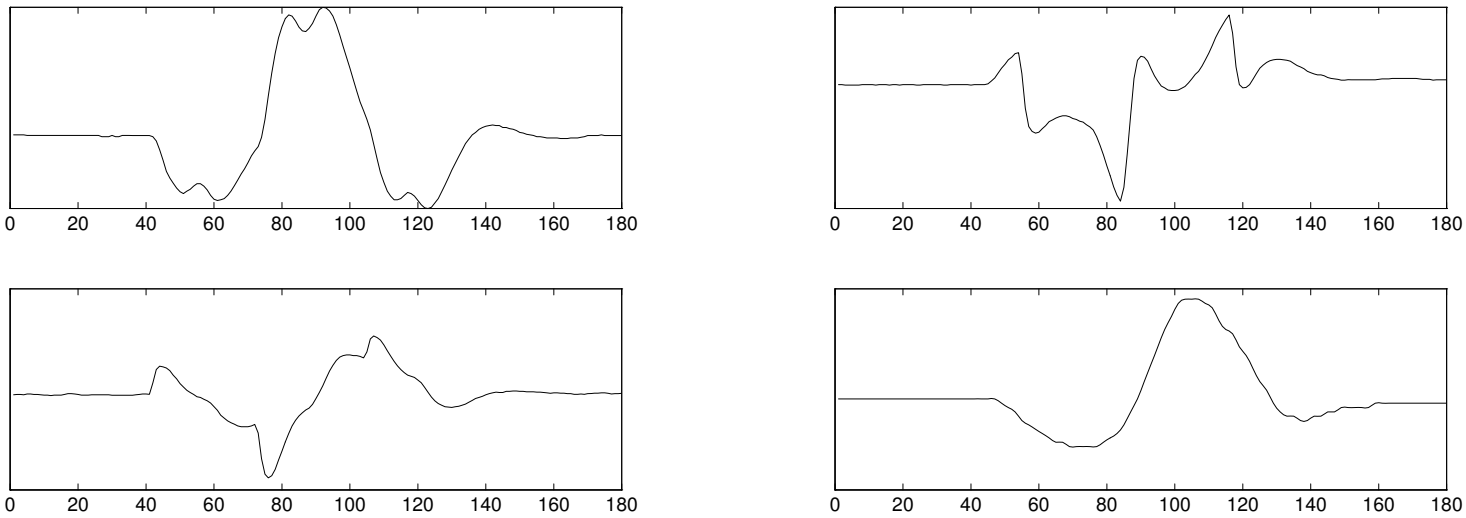
Data from the Gripen Aircraft



Pitch rate, Canard,
Elevator, Leading Edge Flap

- How do the control surface angles affect the pitch rate?

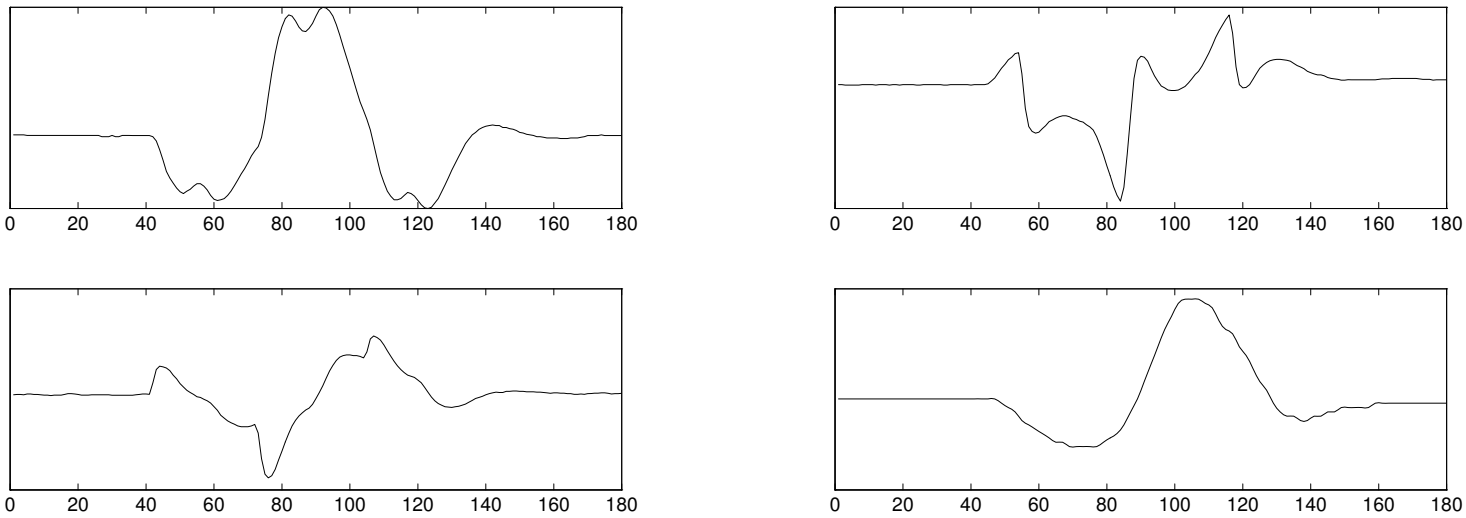
Data from the Gripen Aircraft



Pitch rate, Canard,
Elevator, Leading Edge Flap

- How do the control surface angles affect the pitch rate?
- Simulation of the aircraft

Data from the Gripen Aircraft



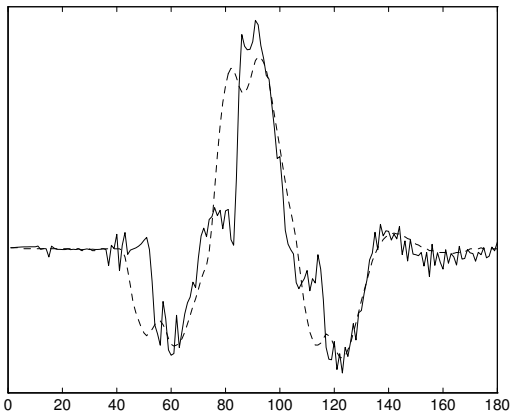
Pitch rate, Canard,
Elevator, Leading Edge Flap

- How do the control surface angles affect the pitch rate?
- Simulation of the aircraft
- Design of autopilot (regulator)

Aircraft Dynamics: From input 1

$y(t)$ pitch rate at time t. $u_1(t)$ canard angle at time t. $T = 1/60$.
Try

$$y(t) = +b_1 u_1(t-T) + b_2 u_1(t-2T) + b_3 u_1(t-3T) + b_4 u_1(t-4T)$$

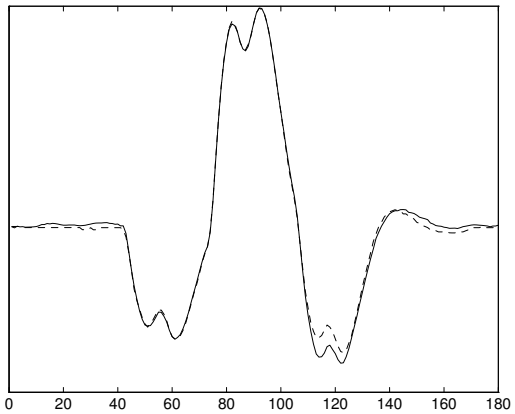


Dashed line: Measured output (Pitch rate). Solid line: Model output, simulated from the fourth order model from canard angle only.

Using All Inputs

u_1 canard angle; u_2 Elevator angle; u_3 Leading edge flap;

$$
\begin{aligned}
y(t) = & -a_1 y(t-T) - a_2 y(t-2T) - a_3 y(t-3T) - a_4 y(t-4T) \\
& + b_1^1 u_1(t-T) + \ldots + b_1^4 u_1(t-4T) \\
& + b_2^1 u_2(t-T) + \ldots + b_1^3 u_3(t-T) + \ldots + b_4^3 u_3(t-4T)
\end{aligned}
$$



Dashed line: Measured output (Pitch rate). Solid line: Model output, simulated from the fourth order model with all three inputs

System Identification: Issues

- Select a class of candidate models
- Select a member in this class using the observed data
- Evaluate the quality of the obtained model
- Design the experiment so that the model will be “good”.

A Typical Problem

Given Observed Input-Output Data: Find a Description of the System that Generated the Data.

System Identification: State-of-the-Art Setup

A Typical Problem

Given Observed Input-Output Data: Find a Description of the System that Generated the Data.

Basic Approach

Find a suitable Model Structure, Estimate its parameters, and compute the response of the resulting model

System Identification: State-of-the-Art Setup

A Typical Problem

Given Observed Input-Output Data: Find a Description of the System that Generated the Data.

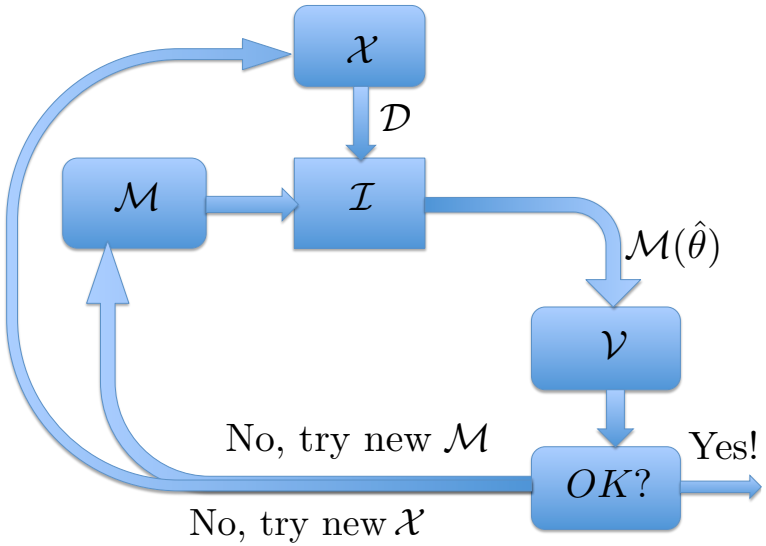
Basic Approach

Find a suitable Model Structure, Estimate its parameters, and compute the response of the resulting model

Techniques

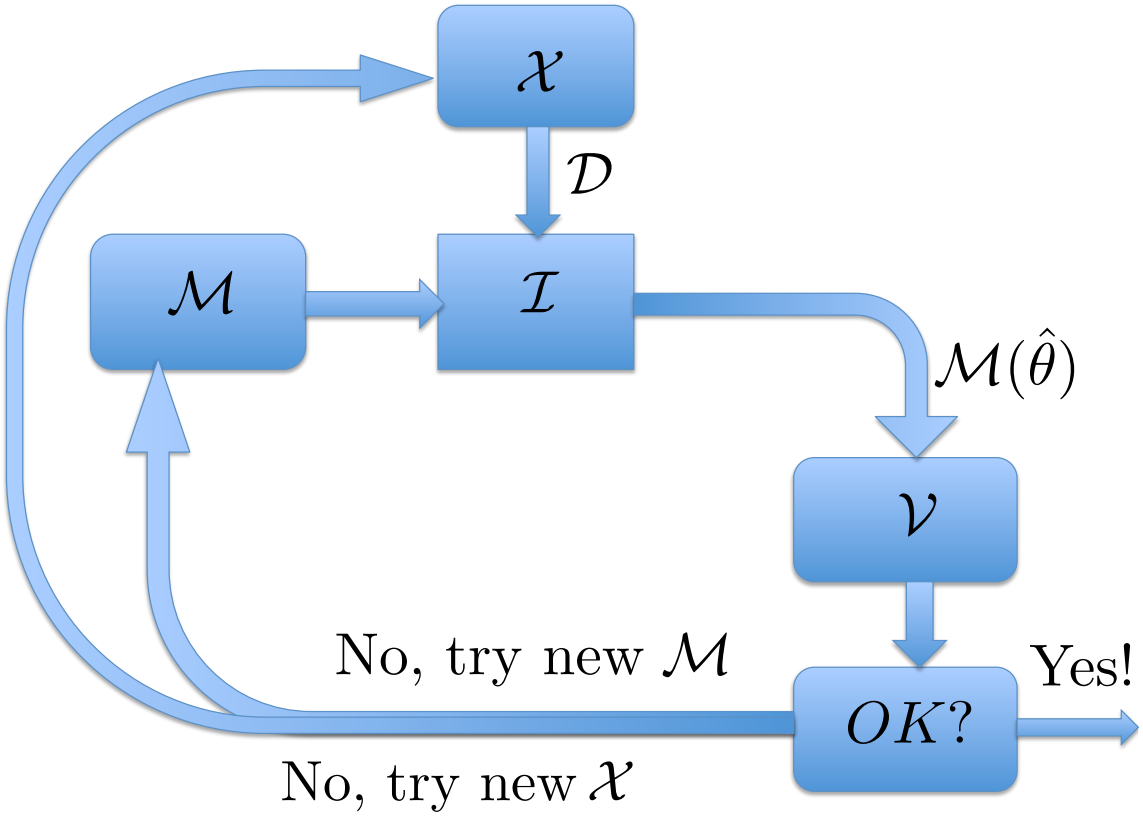
Estimate the parameters by ML techniques/PEM (prediction error methods). Find the model structure by Cross Validation or other validation techniques

The SI Flow



$\mathcal{X}$: The Experiment
$\mathcal{D}$: The Measured Data
$\mathcal{M}$: The Model Set
$\mathcal{I}$: The Identification Method
$\mathcal{V}$: The Validation Procedure

The SI Flow; Model Structures $\mathcal{M}$



Models: General Aspects

- A model is a mathematical expression that describes the connections between measured inputs and outputs, and possibly related noise sequences.

Models: General Aspects

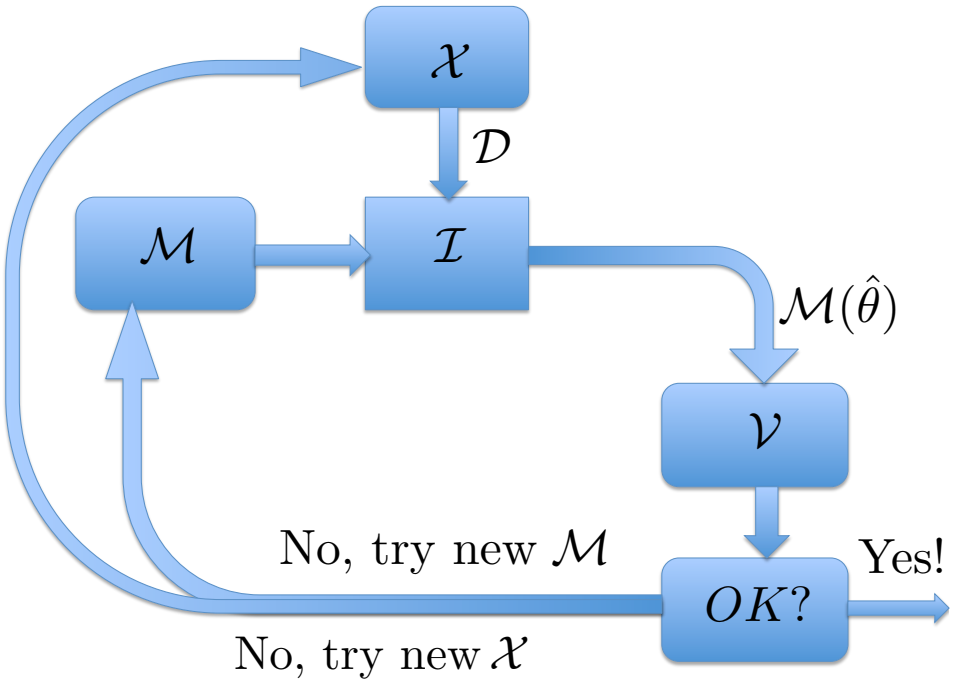
- A model is a mathematical expression that describes the connections between measured inputs and outputs, and possibly related noise sequences.
- They can come in many different forms

- A model is a mathematical expression that describes the connections between measured inputs and outputs, and possibly related noise sequences.
- They can come in many different forms
- Individual models in the structure are labeled with a parameter vector θ

Models: General Aspects

- A model is a mathematical expression that describes the connections between measured inputs and outputs, and possibly related noise sequences.
- They can come in many different forms
- Individual models in the structure are labeled with a parameter vector θ
- A common framework is to describe the model as a predictor of the next output, based on observations of past input-output data.
  Observed input–output (u, y) data up to time t: Z^t
  Model described by predictor: $\mathcal{M}(\theta) : \hat{y}(t|\theta) = g(t, \theta, Z^{t-1})$.

The SI Flow: Estimation: $\mathcal{I}$



A Pragmatic Fit Criterion

If a model, $\hat{y}(t|\theta)$, essentially is a predictor of the next output, is is natural to evaluate its quality by assessing how well it predicts: Form the *Prediction error* and measure its size:

$$\varepsilon(t,\theta) = y(t) - \hat{y}(t|\theta), \quad \ell(\varepsilon(t,\theta))$$

A Pragmatic Fit Criterion

If a model, $\hat{y}(t|\theta)$, essentially is a predictor of the next output, is is natural to evaluate its quality by assessing how well it predicts: Form the *Prediction error* and measure its size:

$$\varepsilon(t,\theta) = y(t) - \hat{y}(t|\theta), \quad \ell(\varepsilon(t,\theta))$$

Typically $\ell(x) = x^2$.

A Pragmatic Fit Criterion

If a model, $\hat{y}(t|\theta)$, essentially is a predictor of the next output, is is natural to evaluate its quality by assessing how well it predicts: Form the *Prediction error* and measure its size:

$$\varepsilon(t,\theta) = y(t) - \hat{y}(t|\theta), \quad \ell(\varepsilon(t,\theta))$$

Typically $\ell(x) = x^2$. How has it performed historically?

$$V_N(\theta) = \sum_{t=1}^{N} \varepsilon^2(t,\theta)$$

A Pragmatic Fit Criterion

If a model, $\hat{y}(t|\theta)$, essentially is a predictor of the next output, is is natural to evaluate its quality by assessing how well it predicts: Form the *Prediction error* and measure its size:

$$\varepsilon(t,\theta) = y(t) - \hat{y}(t|\theta), \quad \ell(\varepsilon(t,\theta))$$

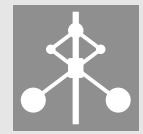
Typically $\ell(x) = x^2$. How has it performed historically?

$$V_N(\theta) = \sum_{t=1}^{N} \varepsilon^2(t,\theta)$$

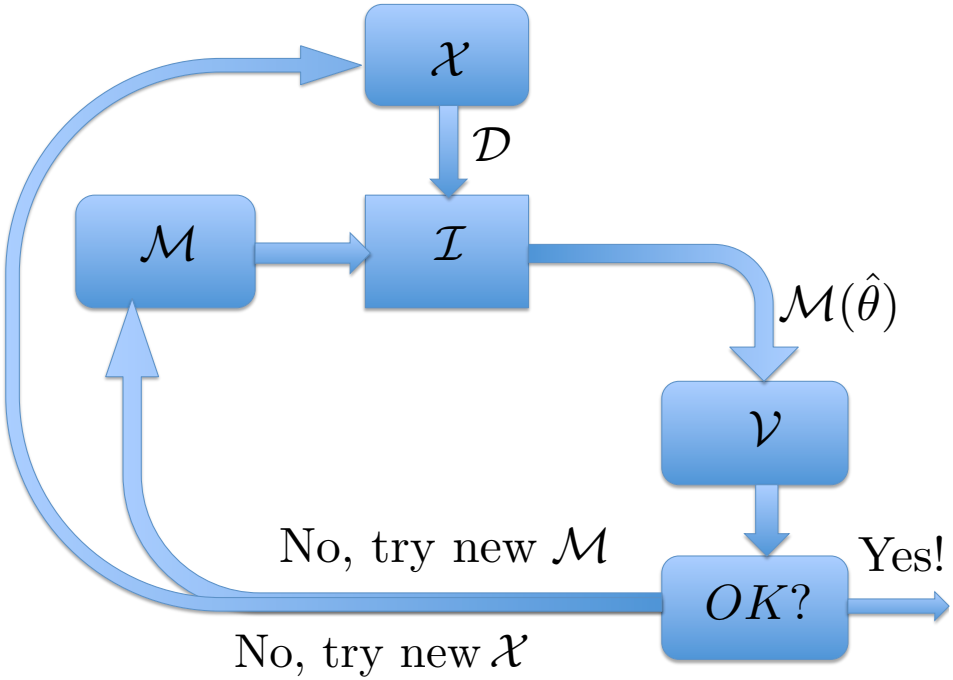
Which model in the structure performed best (Prediction Error Method, PEM)?

$$\hat{\theta}_N = \arg\min_{\theta \in D_{\mathcal{M}}} V_N(\theta)$$

(This is often also the Maximum Likelihood Estimate (MLE).)



Model Estimate Properties $\mathcal{M}(\hat{\theta}_N)$



Model Estimate Properties

As the number of data, N, tends to infinity

- $\hat{\theta}_N \to \theta^* \sim \arg\min_\theta E\varepsilon^2(t,\theta)$
- $\mathcal{M}(\theta^*)$ is the best possible predictor in $\mathcal{M}$

E: Expectation.

Model Estimate Properties

As the number of data, N, tends to infinity

- $\hat{\theta}_N \to \theta^* \sim \arg\min_\theta E\varepsilon^2(t,\theta)$
- $\mathcal{M}(\theta^*)$ is the best possible predictor in $\mathcal{M}$

E: Expectation. This is very nice approximation property:

- The model structure is not large enough: The ML/PEM estimate converges to the best possible approximation of the system.

Model Estimate Properties

As the number of data, N, tends to infinity

- $\hat{\theta}_N \to \theta^* \sim \arg\min_\theta E\varepsilon^2(t,\theta)$
- $\mathcal{M}(\theta^*)$ is the best possible predictor in $\mathcal{M}$

E: Expectation. This is very nice approximation property:

- The model structure is not large enough: The ML/PEM estimate converges to the best possible approximation of the system.
- "Best possible approximation" ...

Model Estimate Properties

As the number of data, N, tends to infinity

- $\hat{\theta}_N \to \theta^* \sim \arg\min_\theta E\varepsilon^2(t,\theta)$
- $\mathcal{M}(\theta^*)$ is the best possible predictor in $\mathcal{M}$

E: Expectation. This is very nice approximation property:

- The model structure is not large enough: The ML/PEM estimate converges to the best possible approximation of the system.
- "Best possible approximation" ...
- ... under the conditions of the experiment

Model Estimate Properties

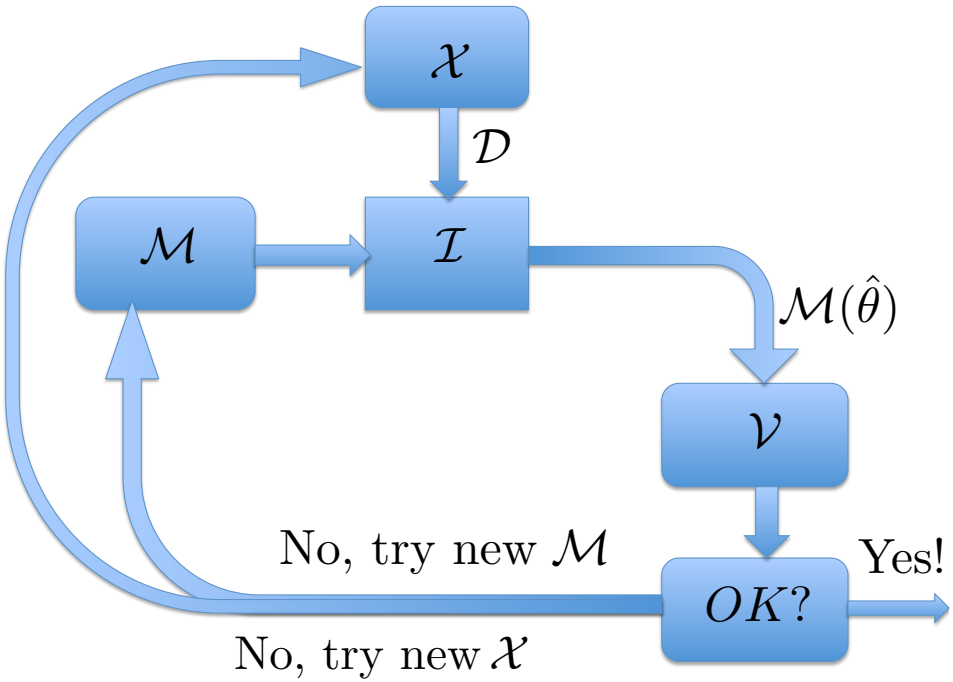
As the number of data, N, tends to infinity

- $\hat{\theta}_N \to \theta^* \sim \arg\min_\theta E\varepsilon^2(t,\theta)$
- $\mathcal{M}(\theta^*)$ is the best possible predictor in $\mathcal{M}$

E: Expectation. This is very nice approximation property:

- The model structure is not large enough: The ML/PEM estimate converges to the best possible approximation of the system.
- "Best possible approximation" ...
- ... under the conditions of the experiment
- (If the model structure is large enough to contain a true description of the system, then the ML/PEM estimated model has (asymptotically) the best accuracy).

Experiment Design: $\mathcal{X}$



$\mathcal{X}$: The design variables: Input, Sampling Interval, Feedback,...
Then we just saw

$$\hat{\theta}_N \to \theta^*(\mathcal{X})$$

$\mathcal{X}$: The design variables: Input, Sampling Interval, Feedback,...
Then we just saw

$$\hat{\theta}_N \to \theta^*(\mathcal{X})$$

- The model $\mathcal{M}(\theta^*(\mathcal{X}))$ is the best approximation of the system under $\mathcal{X}$

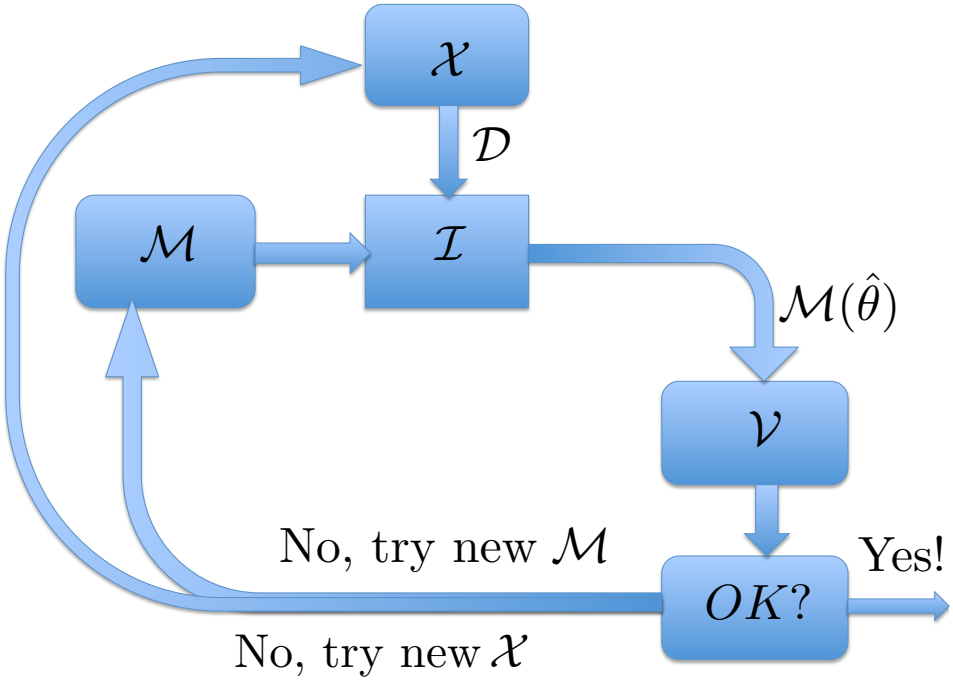
Experiment Design - Basic Principle

$\mathcal{X}$: The design variables: Input, Sampling Interval, Feedback,...
Then we just saw

$$\hat{\theta}_N \to \theta^*(\mathcal{X})$$

- The model $\mathcal{M}(\theta^*(\mathcal{X}))$ is the best approximation of the system under $\mathcal{X}$
- Let the experimental conditions resemble those under which the model is to be used!

Model Validation: $\mathcal{V}$



Model Validation Techniques

"Twist and turn" the model(s) to check if they are good enough for the intended application.

Model Validation Techniques

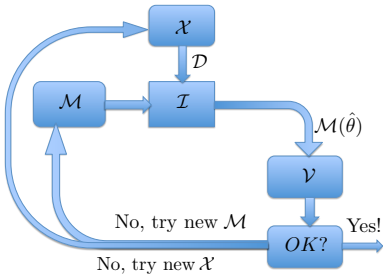
"Twist and turn" the model(s) to check if they are good enough for the intended application.
Essentially a subjective decision. Several basic techniques are available:

1. Simulation (prediction) cross validation
   - Check how well the estimated model can reproduce new, validation data - Can compare different models in that way.

"Twist and turn" the model(s) to check if they are good enough for the intended application.
Essentially a subjective decision. Several basic techniques are available:

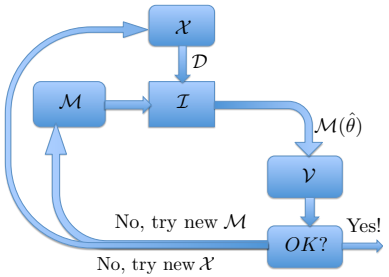
1. Simulation (prediction) cross validation
   - Check how well the estimated model can reproduce new, validation data - Can compare different models in that way.
2. Residual Analysis
   - Are the residuals $\varepsilon(t, \hat{\theta}_N)$ (the "leftovers") unpredictable? They should not be correlated with anything we knew when estimating the model. Check correlation of the residuals with old inputs.

Rest Point



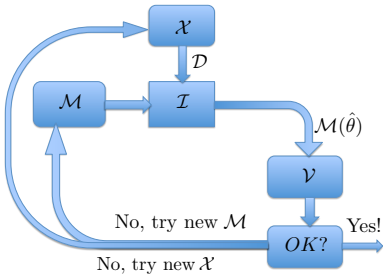
- So this is the System Identification Flow or Loop

Rest Point



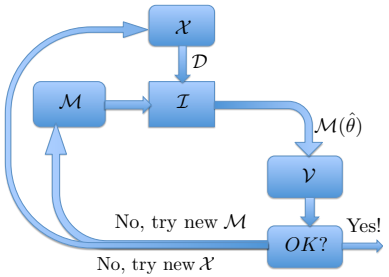
- So this is the System Identification Flow or Loop
- Several essential choices that have to be made, and often revised.

Rest Point



- So this is the System Identification Flow or Loop
- Several essential choices that have to be made, and often revised.
- Many of the choices have to be taken with the intended model use in mind and thus have a subjective flavour.

Rest Point



- So this is the System Identification Flow or Loop
- Several essential choices that have to be made, and often revised.
- Many of the choices have to be taken with the intended model use in mind and thus have a subjective flavour.
- Let us now turn to some specific illustrations for linear models (Requires some more mathematical background.)

More Technical Details: Linear Models

$$y(t) = G(q,\theta)u(t) + v(t); \quad G(q,\theta)u(t) = \sum_{k=1}^{\infty} g_k u(t-k),$$

More Technical Details: Linear Models

$$y(t) = G(q,\theta)u(t) + v(t); \quad G(q,\theta)u(t) = \sum_{k=1}^{\infty} g_k u(t-k),$$

$$v \text{ has spectrum } \Phi_v(\omega) = \lambda |H(e^{i\omega},\theta)|^2$$

$$v(t) = H(q,\theta)e(t) \quad e(t) \text{ white noise}$$

More Technical Details: Linear Models

$$y(t) = G(q,\theta)u(t) + v(t); \quad G(q,\theta)u(t) = \sum_{k=1}^{\infty} g_k u(t-k),$$

$$v \text{ has spectrum } \Phi_v(\omega) = \lambda |H(e^{i\omega},\theta)|^2$$

$$v(t) = H(q,\theta)e(t) \quad e(t) \text{ white noise}$$

General Description of a Linear Model

$$y(t) = G(q,\theta)u(t) + H(q,\theta)e(t)$$

More Technical Details: Linear Models

$$y(t) = G(q,\theta)u(t) + v(t); \quad G(q,\theta)u(t) = \sum_{k=1}^{\infty} g_k u(t-k),$$

$$v \text{ has spectrum } \Phi_v(\omega) = \lambda |H(e^{i\omega},\theta)|^2$$
$$v(t) = H(q,\theta)e(t) \quad e(t) \text{ white noise}$$

General Description of a Linear Model

$$y(t) = G(q,\theta)u(t) + H(q,\theta)e(t)$$

The Prediction Errors

$$\varepsilon(t,\theta) = H^{-1}(q,\theta)[y(t) - G(q,\theta)u(t)]$$

Suppose the data is generated by a true linear system $G_0(q)$ and that the prediction errors are pre-filtered by a filter $L(q)$,

$$\hat{\theta}_N = \arg\min \sum (L(q)\varepsilon(t,\theta))^2$$

Suppose the data is generated by a true linear system $G_0(q)$ and that the prediction errors are pre-filtered by a filter $L(q)$,

$$\hat{\theta}_N = \arg\min \sum (L(q)\varepsilon(t,\theta))^2$$

Then

$$\hat{\theta}_N \to \theta^* = \arg\min_\theta \int_{-\pi}^{\pi} |G(e^{i\omega},\theta) - G_0(e^{i\omega})|^2 Q(\omega) d\omega$$

$$Q(\omega) = \frac{|L(e^{i\omega})|^2 \Phi_u(\omega)}{|H(e^{i\omega},\theta)|^2}$$

Asymptotic Properties for Linear Models

Suppose the data is generated by a true linear system $G_0(q)$ and that the prediction errors are pre-filtered by a filter $L(q)$,

$$\hat{\theta}_N = \arg\min \sum (L(q)\varepsilon(t,\theta))^2$$

Then

$$\hat{\theta}_N \to \theta^* = \arg\min_\theta \int_{-\pi}^{\pi} |G(e^{i\omega},\theta) - G_0(e^{i\omega})|^2 Q(\omega) d\omega$$

$$Q(\omega) = \frac{|L(e^{i\omega})|^2 \Phi_u(\omega)}{|H(e^{i\omega},\theta)|^2}$$

So the resulting model is closest to the true system in a norm defined by Q.

Note that we can affect $Q(\omega) = \frac{|L(e^{i\omega})|^2 \Phi_u(\omega)}{|H(e^{i\omega},\theta)|^2}$ by choosing $L - -(\mathcal{I})$, the input spectrum $\Phi_u - -(\mathcal{X})$ and the noise model $H - -(\mathcal{M})$, so it depends on all boxes in the identification chart.

Note that we can affect $Q(\omega) = \frac{|L(e^{i\omega})|^2 \Phi_u(\omega)}{|H(e^{i\omega},\theta)|^2}$ by choosing $L - -(\mathcal{I})$, the input spectrum $\Phi_u - -(\mathcal{X})$ and the noise model $H - -(\mathcal{M})$, so it depends on all boxes in the identification chart.

- Identification for simulation with an input u_* $\Rightarrow$ Make Q equal to the spectrum of u_*

Note that we can affect $Q(\omega) = \frac{|L(e^{i\omega})|^2 \Phi_u(\omega)}{|H(e^{i\omega},\theta)|^2}$ by choosing $L - -(\mathcal{I})$, the input spectrum $\Phi_u - -(\mathcal{X})$ and the noise model $H - -(\mathcal{M})$, so it depends on all boxes in the identification chart.

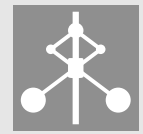
- Identification for simulation with an input u_* $\Rightarrow$ Make Q equal to the spectrum of u_*
- Identification for control $\Rightarrow$ Make Q large at the intended cross-over frequency ($\approx$ the intended bandwidth)

Conclusions

- Identification is a work-flow loop with nodes that contain essential user choices

Conclusions

- Identification is a work-flow loop with nodes that contain essential user choices
- These may (should) depend on the intended use of the final model



Conclusions

- Identification is a work-flow loop with nodes that contain essential user choices
- These may (should) depend on the intended use of the final model
- Concrete illustration for linear system models