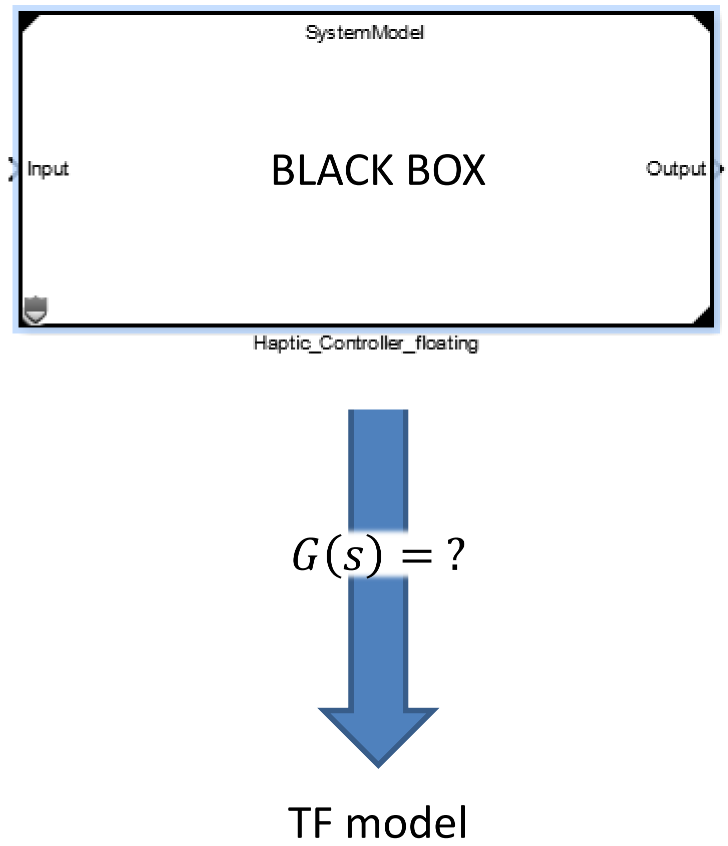
Transfer function identification with Matlab

Mara Tanelli

GOALS

1. Find a TF model for the system
2. Validate the model
3. (Use the model for control purposes to design an LQ controller → later!)



Identification with generic inputs

function
tfest(data,n°poles,n°zeros,delay)

Finds a TF with a generic input
In general, it works better with sinusoidal inputs

MATLAB

System Identification Toolbox

Identification with generic inputs

function
tfest(data,n°poles,n°zeros,delay)

Iddata object

Create an *iddata object* to use for the function *tfest*

```
datainout = iddata(output,input,Ts);
```

Output of the system response

Input of the system response

Sampling time

Identification with generic inputs

function
tfest(data,n°poles,n°zeros,delay)

finddelay(X,Y)

Find the number of samples of delay between Y and X

```
iodelay = finddelay(input,output)*Ts;
```

function
tfest(data,n°poles,n°zeros,delay)

```
%% TF Identification

iodelay = finddelay(input,output)*Ts;
datainout = iddata(output,input,Ts);
n_poles = 1;
n_zeros = 0;
TF = tfest(datainout,n_poles,n_zeros,iodelay);
```

TF is an *idtf object*

We try to identify a simple 1-pole TF

Identification with step inputs

function
tfest(data,n°poles,n°zeros,delay

```
%% TF Identification

iodelay = finddelay(input,output)*Ts;
datainout = iddata(output,input,Ts);
n_poles = 1;
n_zeros = 0;
TF = tfest(datainout,n_poles,n_zeros,iodelay);
```

```
num = TF.Numerator;
den = TF.Denominator;

G2 = tf(num,den);
```

With the '.' operator, we can extract NUM and DEN from the TF object

Then use **tf** to create the usual transfer function

function
tfest(data,n°poles,n°zeros,delay)

```
%% TF Identification

iodelay = finddelay(input,output)*Ts;
datainout = iddata(output,input,Ts);
n_poles = 1;
n_zeros = 0;
TF = tfest(datainout,n_poles,n_zeros,iodelay);
```

```
num = TF.Numerator;
den = TF.Denominator;

G2 = tf(num,den);

s = tf('s');
G1 = 1.57*exp(-0.8*s)/(1+3.7*s);
```

$$G_2(s) = \frac{0.374}{s + 0.3169}$$

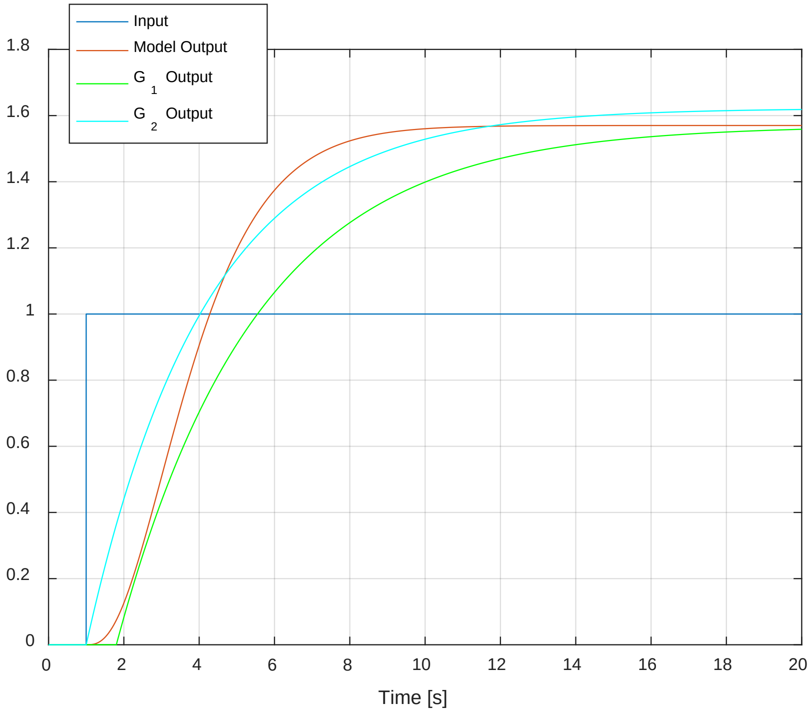
$$G_1(s) = \frac{e^{-0.8s}}{1 + 3.7s}$$

Identification with step inputs

Function
lsim(sistema,ingresso,tempo)

Simulates a dynamical system

```
output_G1 = lsim(G1,input,time);
output_G2 = lsim(G2,input,time);

figure
plot(time,input)
hold on
plot(time,output)
xlabel('Time [s]')
grid
plot(time,output_G1,'g')
plot(time,output_G2,'c')
legend('Input','Model Output','G_1 Output','G_2 Output')
xlim([0 sim_time])
```



Identification with step inputs

- Linearity check

Check that, if you vary the input amplitude, the output one varies proportionally

In an LTI system, if $y_1(t)$ and $y_2(t)$ are the outputs corresponding to $u_1(t)$ and $u_2(t)$, respectively, and $u_2(t) = ku_1(t)$, then $y_2(t) = ky_1(t)$

Identification with step inputs

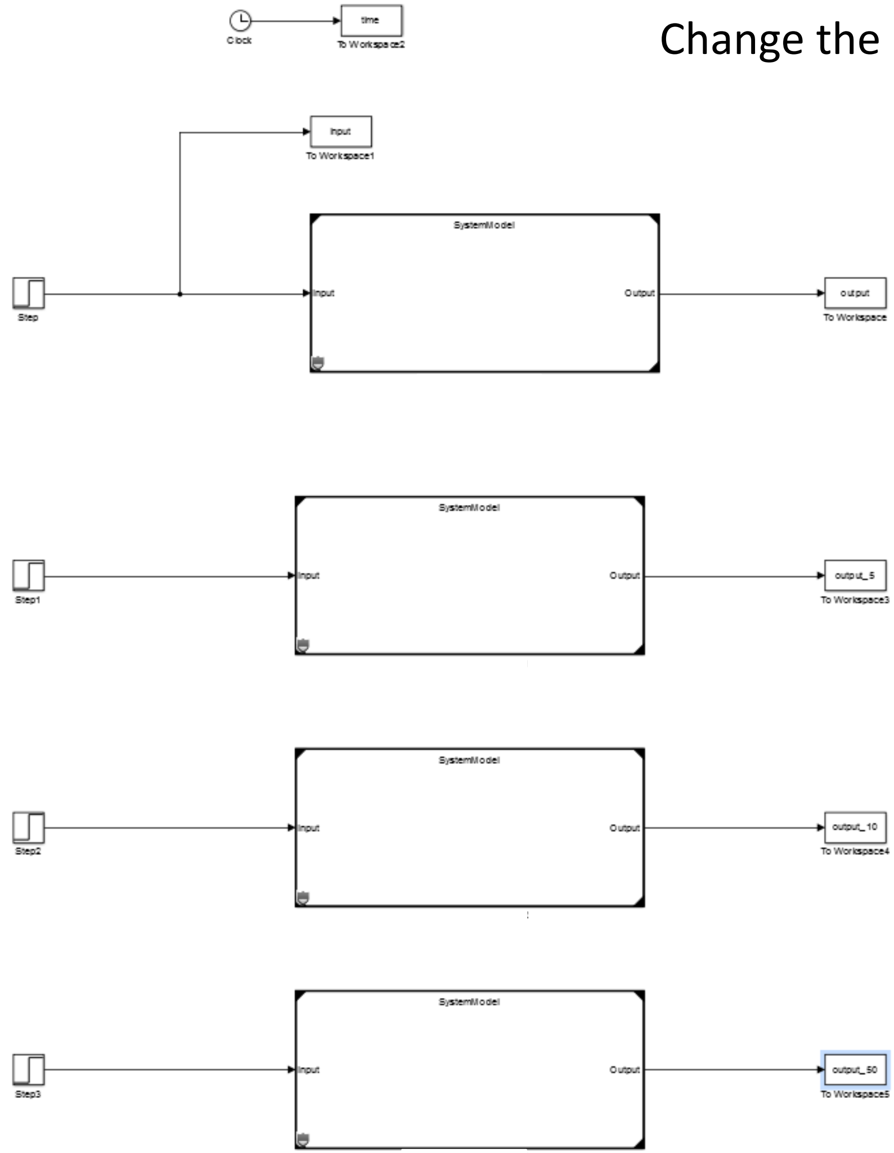
Change the input amplitudes

$A = 1$

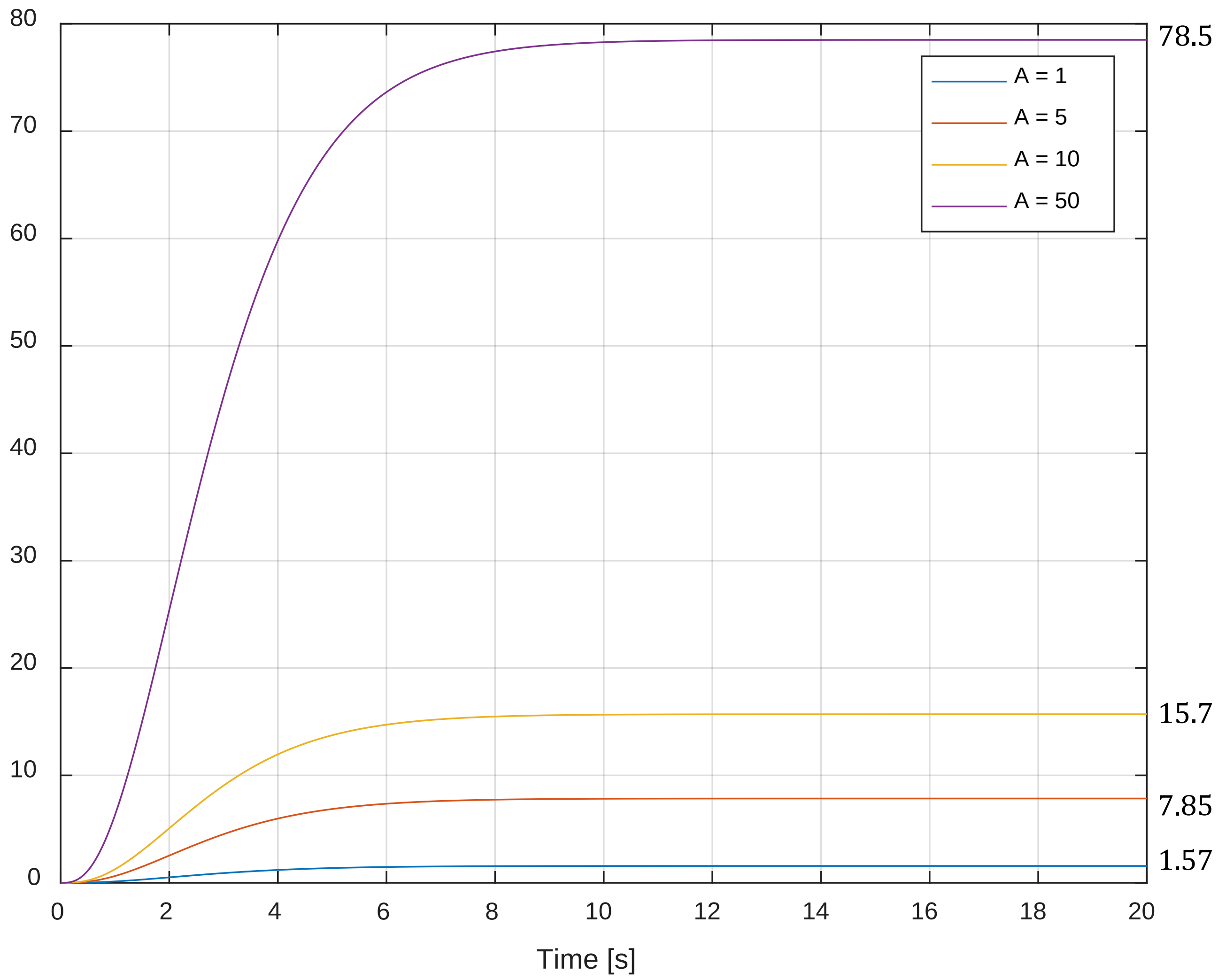
$A = 5$

$A = 10$

$A = 50$

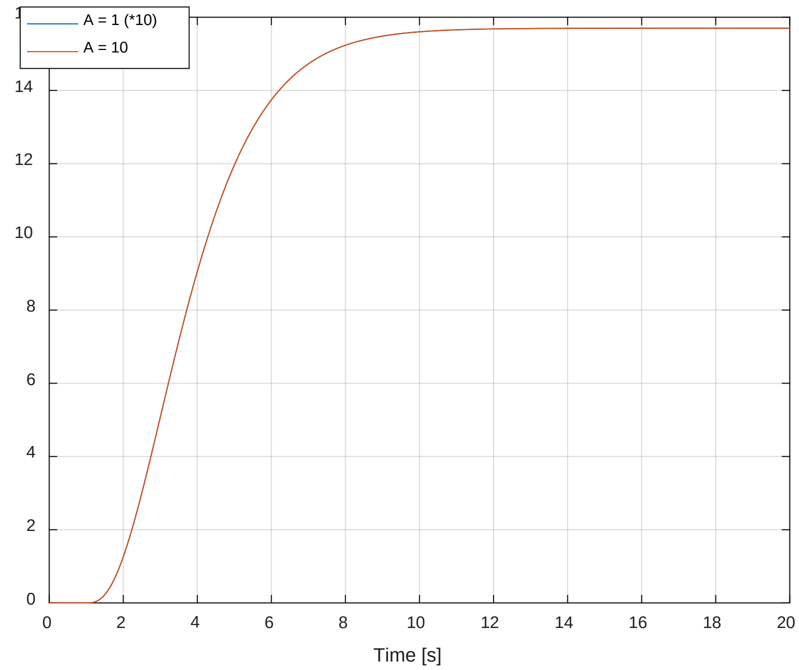


Identification with step inputs



The output for $A = 10$ is 10 times the output obtained with $A = 1$

```
figure
plot(time,output*10)
hold on
plot(time,output_10)
legend('A = 1 (*10)','A = 10')
xlabel('Time [s]')
grid
```



Frequency response theorem

If we apply, to an asympt. stable LTI system with TF $G(s)$ the sinusoidal input

$$u(t) = U \sin(\omega_0 t)$$

The steady-state output has the form

$$\tilde{y}(t) = |G(j\omega_0)| U \sin(\omega_0 t + \arg G(j\omega_0))$$

for all initial conditions.

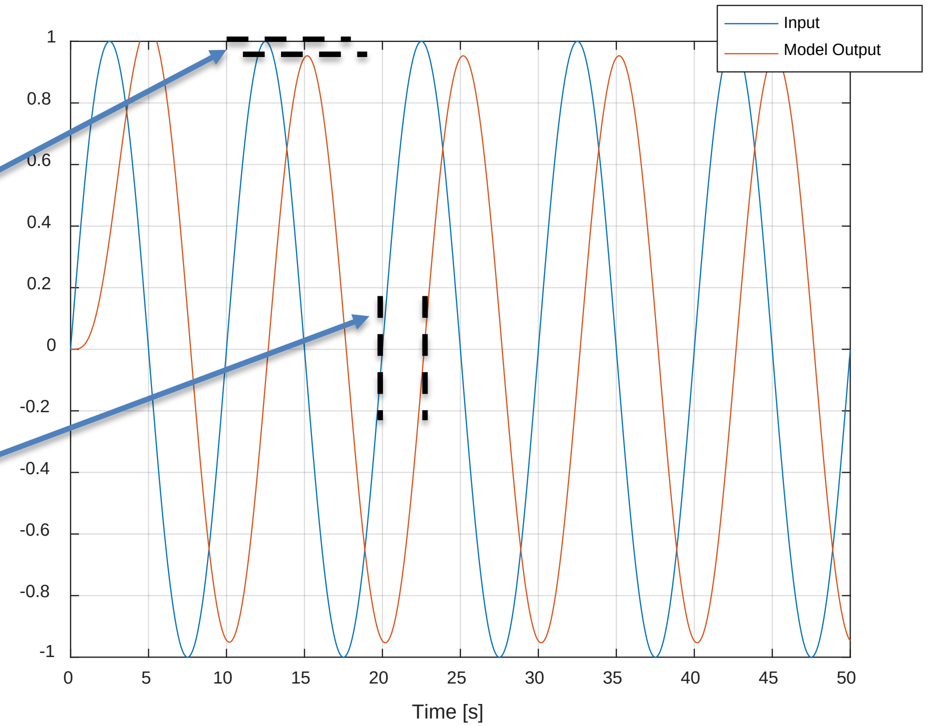
- **Single sinusoids** → We find $G(j\omega)$ <u>pointwise</u>

Different tests with single sinusoidal inputs

$$u(t) = U\sin(\omega_0 t)$$

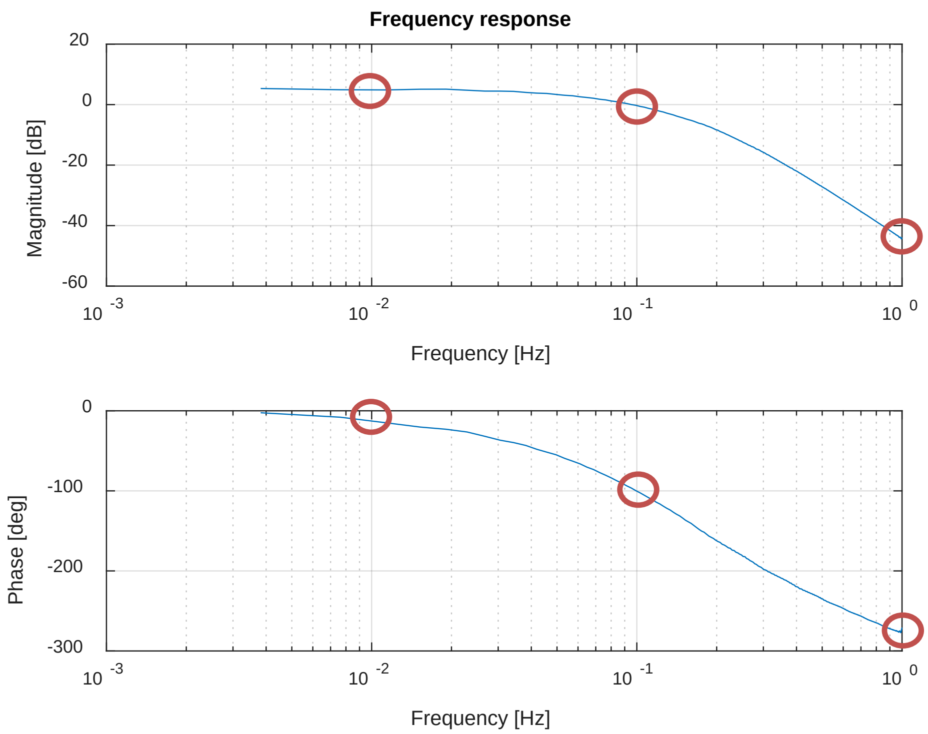
Record each output

$$\tilde{y}(t) = |G(j\omega_0)|U\sin(\omega_0 t + \arg G(j\omega_0))$$



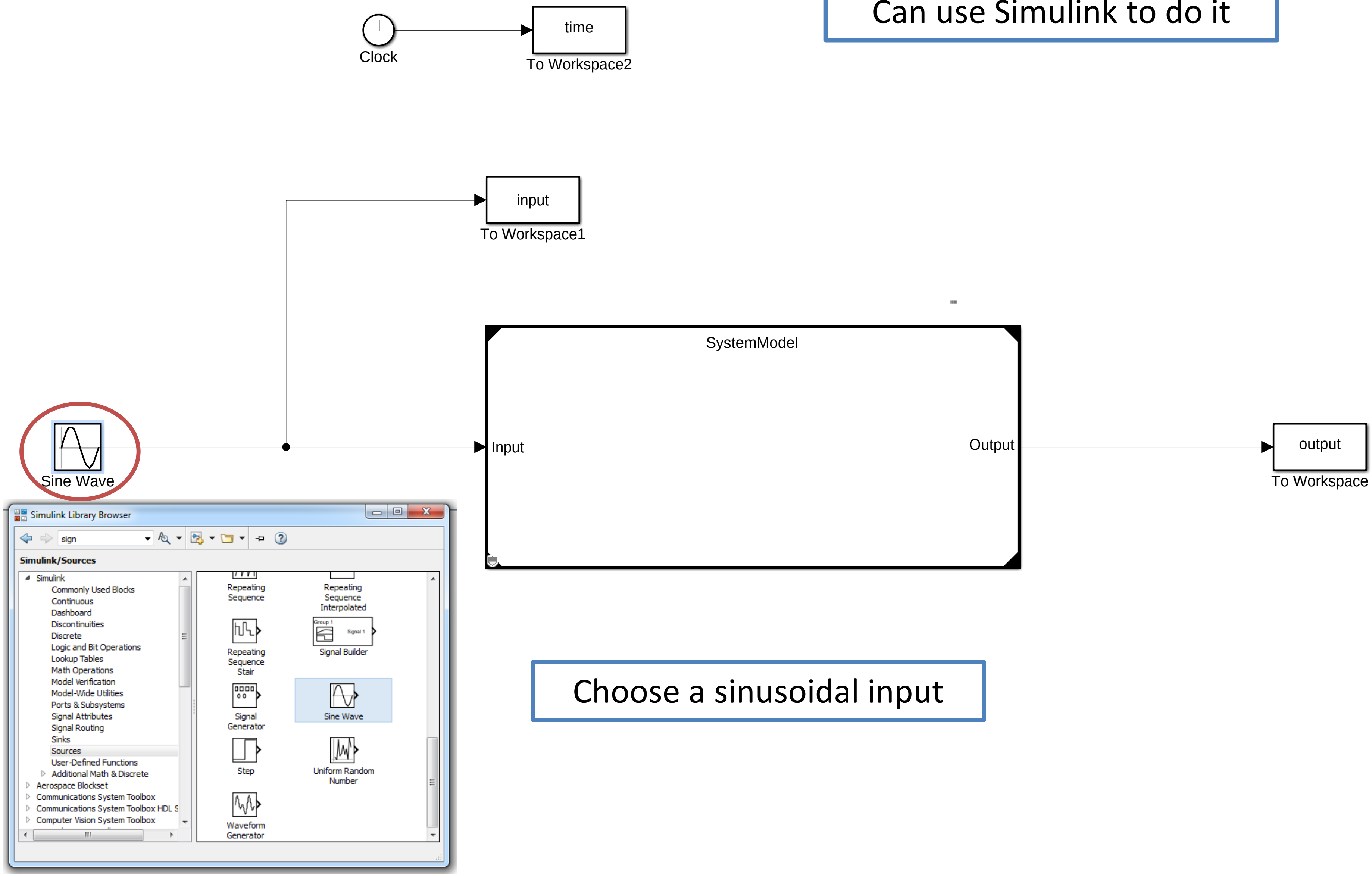
- **Single sinusoids**

Find an estimate of $G(j\omega)$ <u>pointwise with a sufficient number of tests</u>



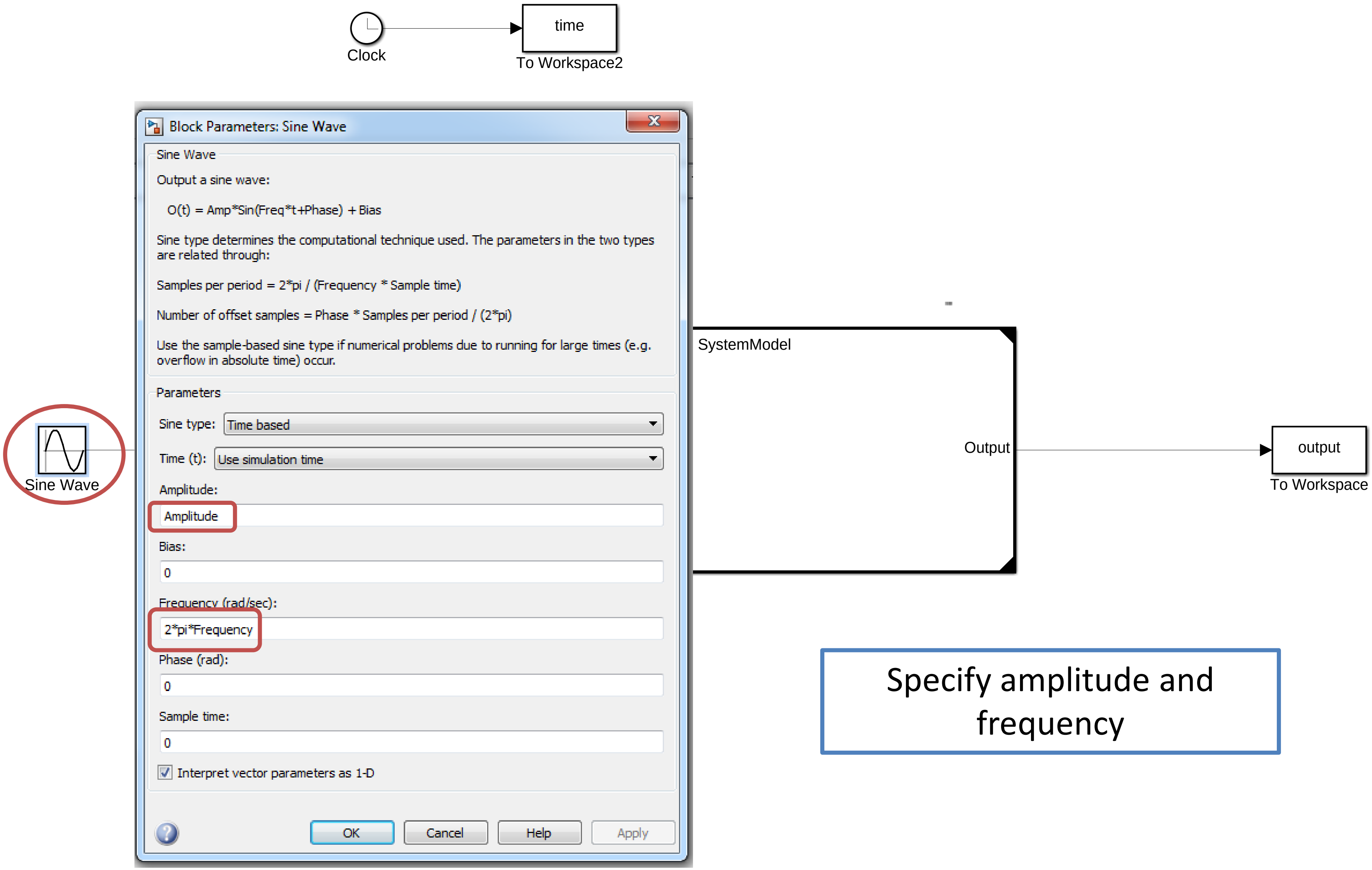
Identification with the frequency response theorem

Can use Simulink to do it



Choose a sinusoidal input

Identification with the frequency response theorem



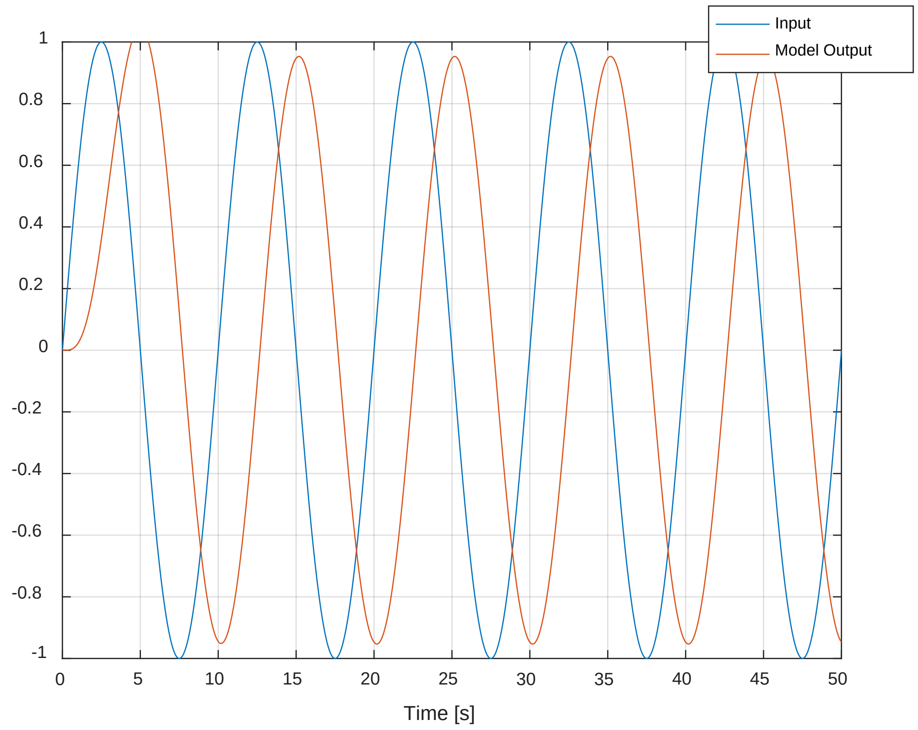
Specify amplitude and frequency

We choose:

$$u(t) = U \sin(\omega_0 t)$$

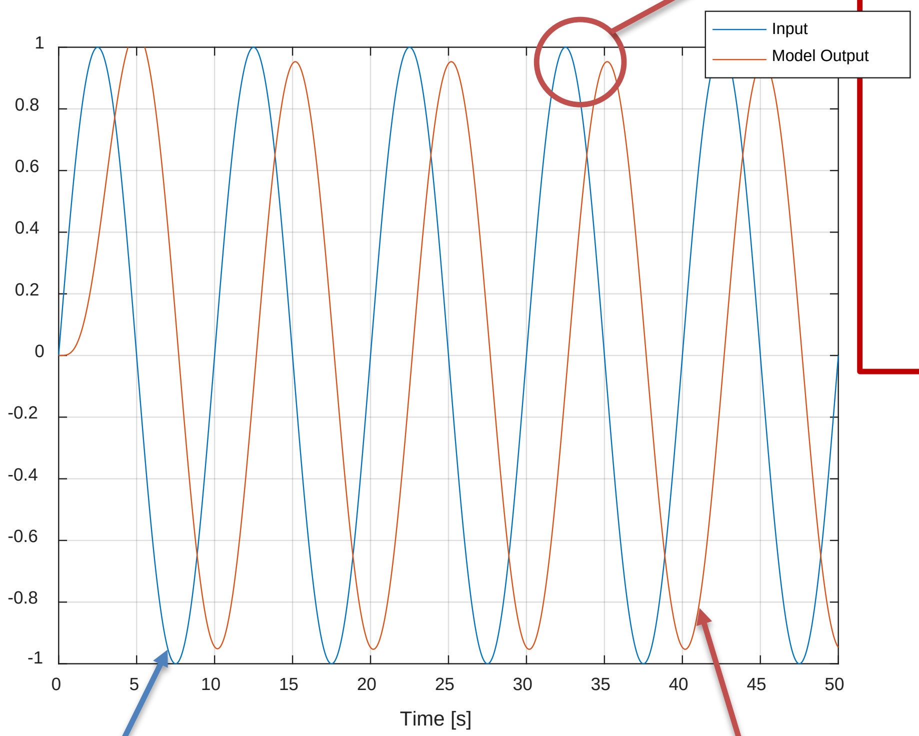
$$U = 1$$
$$\omega_0 = 2\pi * 0.1\ Hz$$

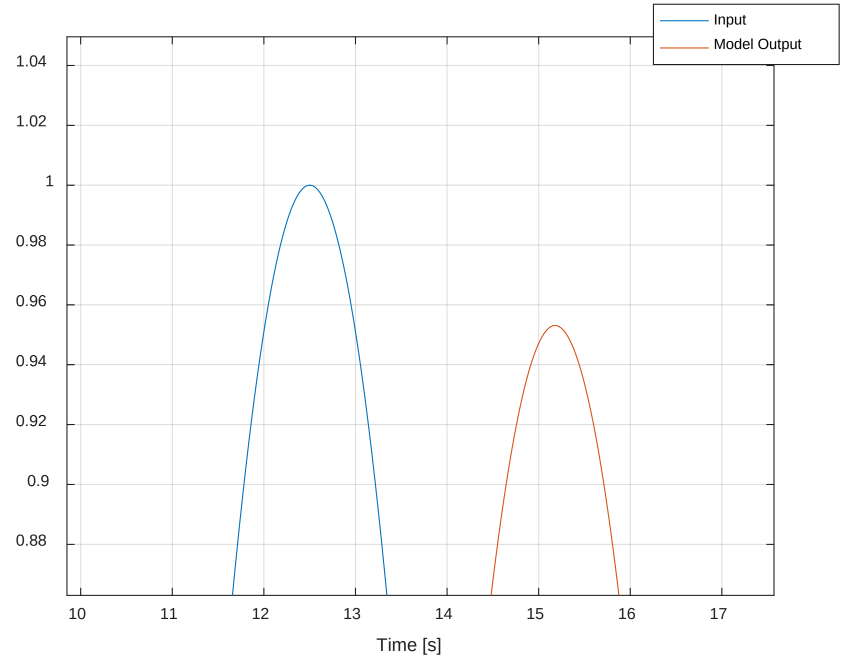
```
%% Parameters

Ts = 1e-3;

Amplitude = 1;

Frequency = 0.1;

sim_time = 50;

%% Simulation

sim BlackBox_Model_SingoleSinusoidi.slx

%% Plots

figure
plot(time,input)
hold on
plot(time,output)
legend('Input','Model Output')
xlabel('Time [s]')
grid
ylim([-1 1])
```



Identification with the frequency response theorem

$$\omega_0 = 2\pi * 0.1Hz$$



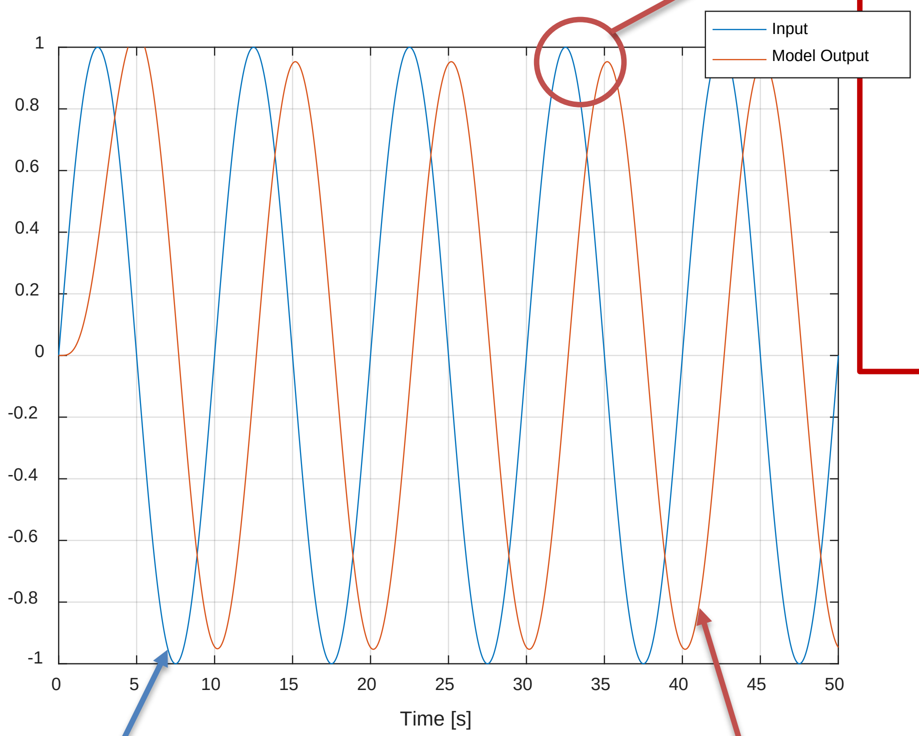


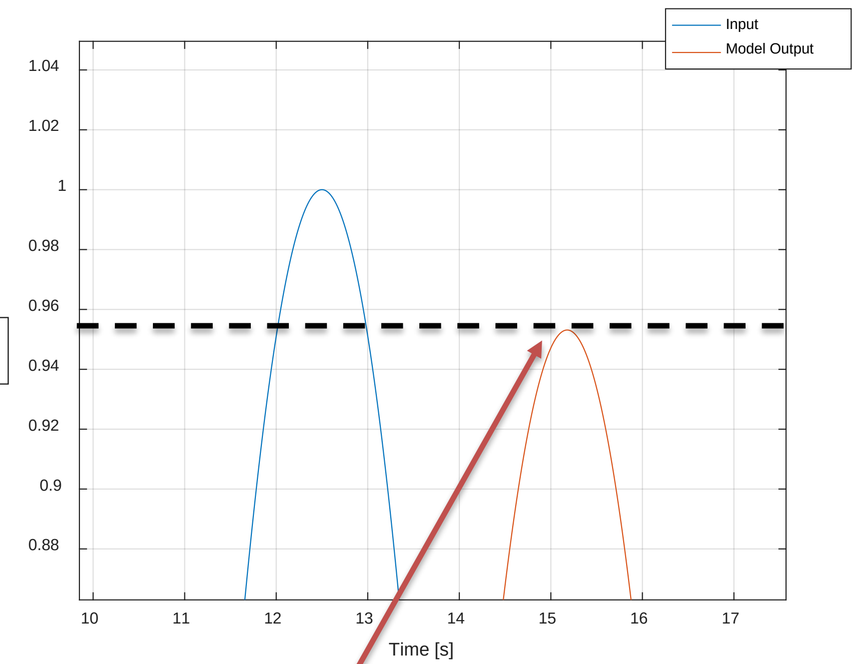
find $|G(j\omega_0)|$ graphically

$$u(t) = U\sin(\omega_0 t)$$

$$\tilde{y}(t) = |G(j\omega_0)|U\sin(\omega_0 t + \arg G(j\omega_0))$$

$$\omega_0 = 2\pi * 0.1Hz$$





$$|G(j\omega_0)|U \cong 0.953$$

$$U = 1$$

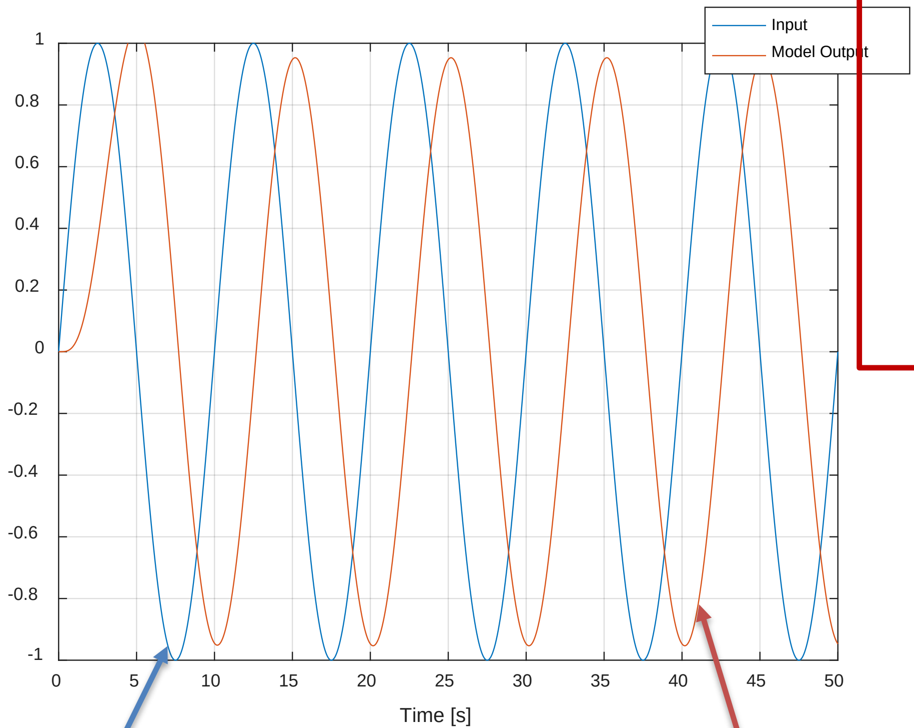
$$|G(j\omega_0)| \cong 0.953$$

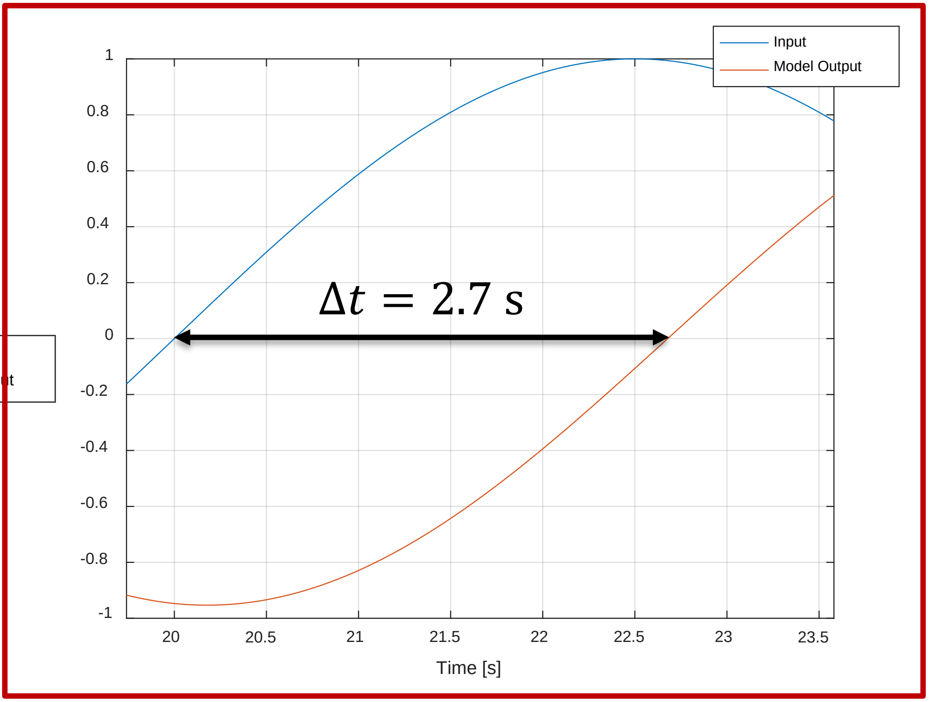
$$u(t) = U\sin(\omega_0 t)$$

$$\tilde{y}(t) = |G(j\omega_0)|U \sin(\omega_0 t + \arg G(j\omega_0))$$

Identification with the frequency response theorem

$$\omega_0 = 2\pi * 0.1Hz$$





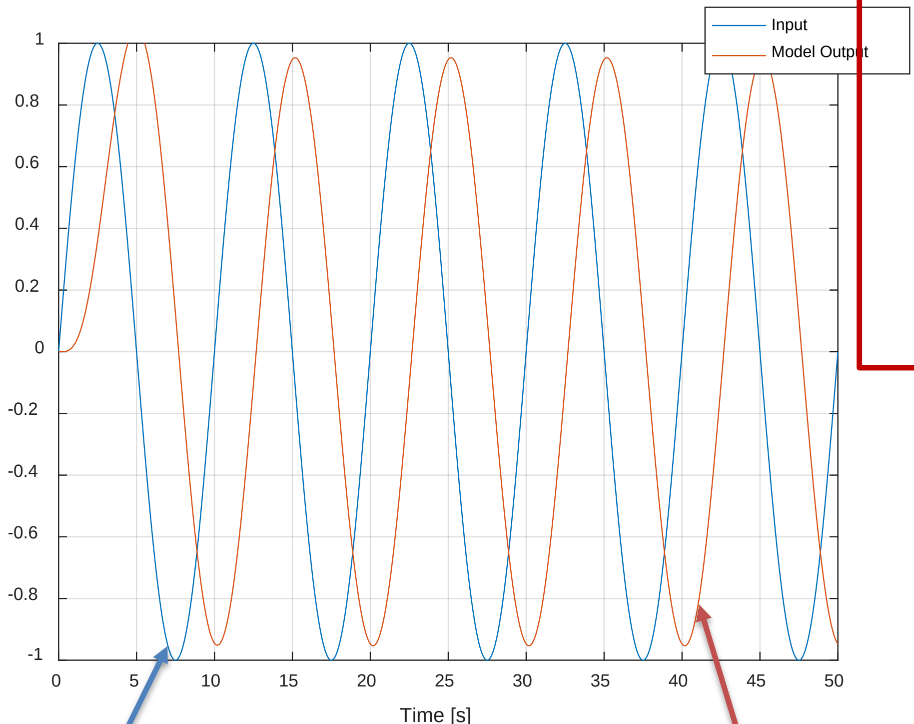
Find $\arg G(j\omega_0)$

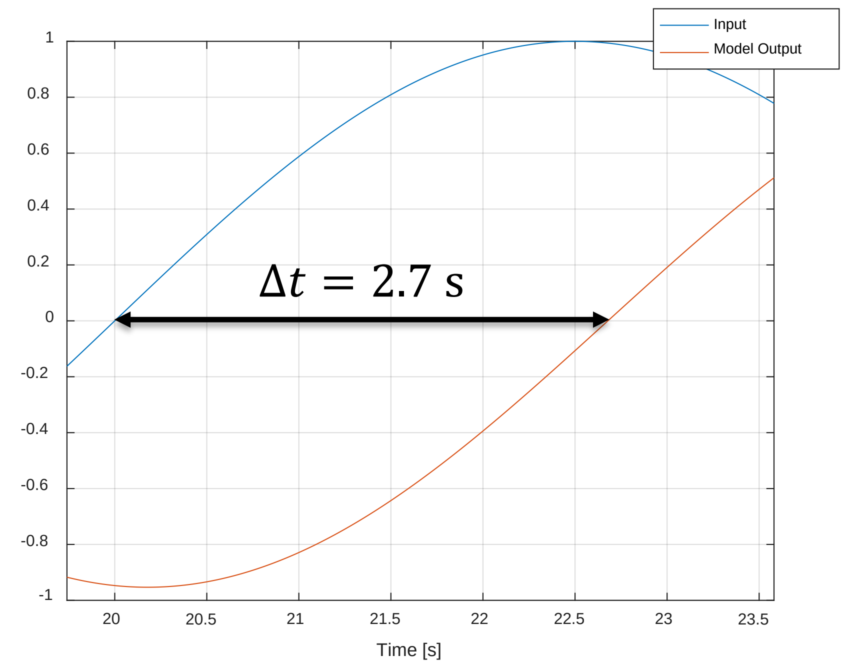
$$u(t) = U\sin(\omega_0 t)$$

$$\tilde{y}(t) = |G(j\omega_0)|U\sin(\omega_0 t + \arg G(j\omega_0))$$

Identification with the frequency response theorem

$$\omega_0 = 2\pi * 0.1Hz$$





$$\arg G(j\omega_0) = -\omega_0 \Delta t \frac{180}{\pi} \cong -97.2\,°$$

$$u(t) = U \sin(\omega_0 t)$$

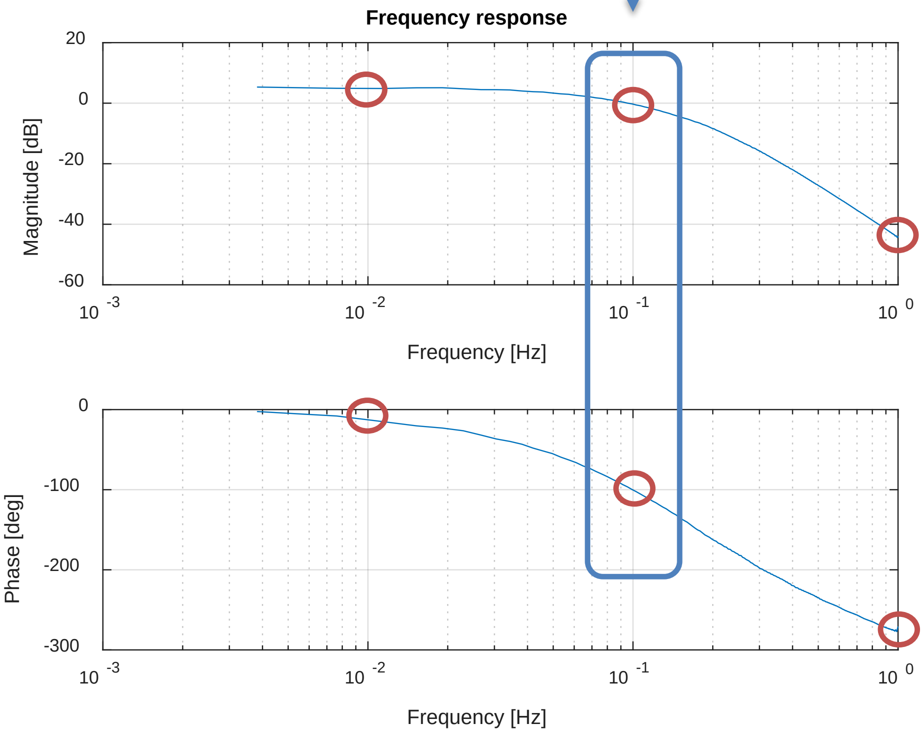
$$\tilde{y}(t) = |G(j\omega_0)| U \sin(\omega_0 t + \arg G(j\omega_0))$$

- **Single sinusoids**

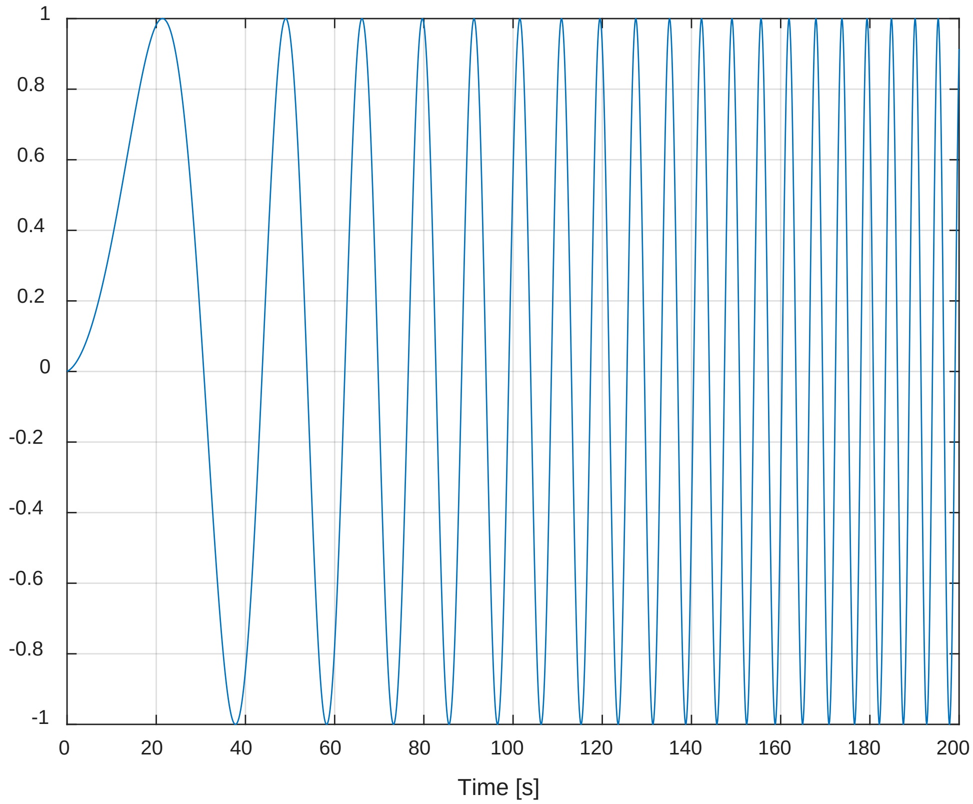
The approach can be not so convenient if many points are needed

The **frequency sweep** can be easier to use

Obtained estimate



- **Frequency Sweep** → Use an input with given amplitude a linearly varying frequency



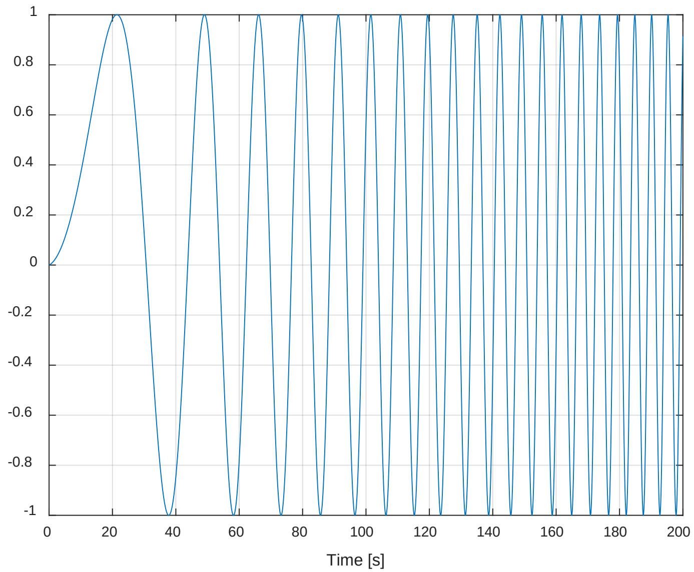
$$\boxed{y(t) = A\sin(\phi(t))}$$

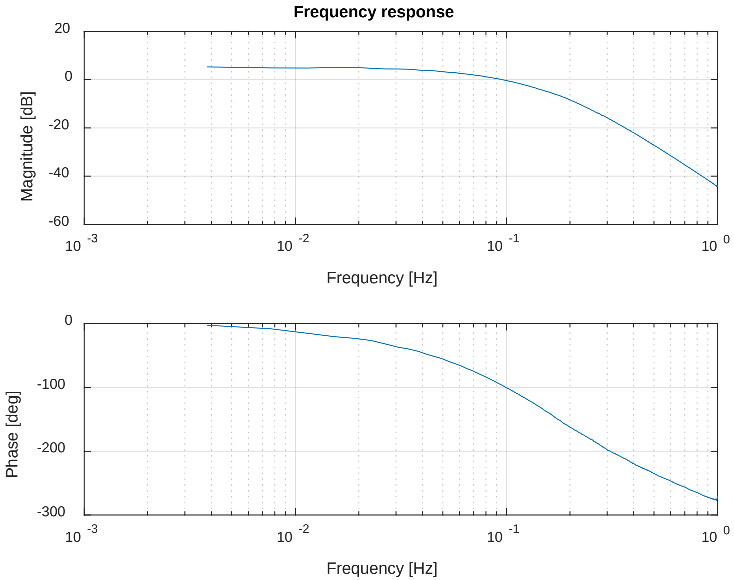
$$\phi(t) = \phi_0 + 2\pi \int_0^t f(\tau)\, d\tau$$

$$\phi(t) = \phi_0 + 2\pi \int_0^t (f_0 + k\tau)\, d\tau$$

$$\boxed{y(t) = A\sin\left[\phi_0 + 2\pi\left(f_0 t + \frac{k}{2} t^2\right)\right]}$$

- **Frequency Sweep**





The frequency should vary slowly, to move the system around steady-state conditions

now we get the whole $G(j\omega)$ in the considered frequency range