Intuitive understanding of superposition and time invariance of **linear** dynamic systems: sequence of steps for a given target behaviour (example)

Video presentation and materials at http://personales.upv.es/asala/YT/V/linregla3EN.html

This code executed in Matlab **R2022b** (Linux)

Objectives: intuitively understand the concept of a linear time-invariant (dynamic) system, and how to use "experimental" time response to compute input profiles that achieve certain objectives.

Note: rigorously speaking, the ideas here only apply "exactly" to systems without "inertia" (that is, 1st order), see bottom note. Theoretical details out of the scope of this introductory material.

Table of Contents

Step response test to gather data

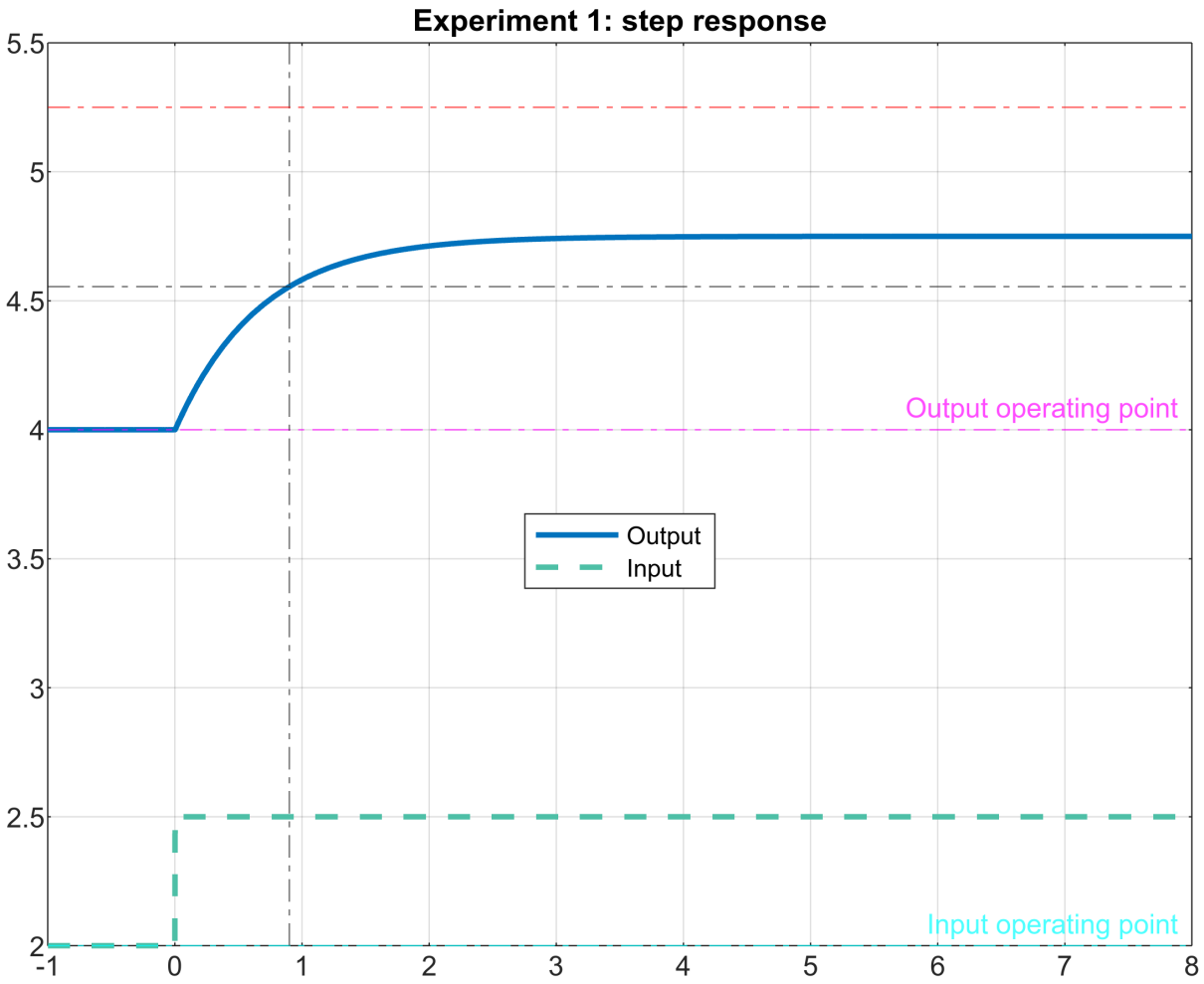
Let us consider a certain unknown system, of which we only know that it is linear (or approximately) around an operating point u=2, y=4;

Let's simulate its step response of 0.5 (incremental)

```
u_op=2; y_op=4;
inc_u=0.5;
u=@(t) u_op+inc_u*(t>=0);
Y=simulsystem(u); %code at the end... we will intentionally NOT see it.
yline(y_op,'-.m',Label="Output operating point")
yline(u_op,'-.c',Label="Input operating point")
```

We will highlight on the graph certain lines that we will need in later developments.

```
desired_settling_time=0.9;
xline(desired_settling_time,'-.')
yline(5.25,'-.r')
yline(4.555,'-.')
legend("Output","Input",Location="best"), title("Experiment 1: step response")
```




```
Final_value=Y(end) %final equilibrium value
```

```
Final_value = 4.7500
```

Problems involving computation of input step amplitude

With only the above information from the step test, we can answer several questions.

Prefixed final value

1.) What input will be needed to raise the output to **5.25** units?

From linearity (i.e., "proportionality"), if with an input increase of 0.5 it goes up 0.75 units, to go up to 5.25 we need:

```
inc_output=Final_value-y_op %experiment result
```

```
inc_output = 0.7500
```

```
static_gain=inc_output/inc_u %increment per unit input
```

```
static_gain = 1.5000
```

```
desired_inc_output=5.25-y_op
```

```
desired_inc_output = 1.2500
```

```
computed_inc_input=desired_inc_output/static_gain
```

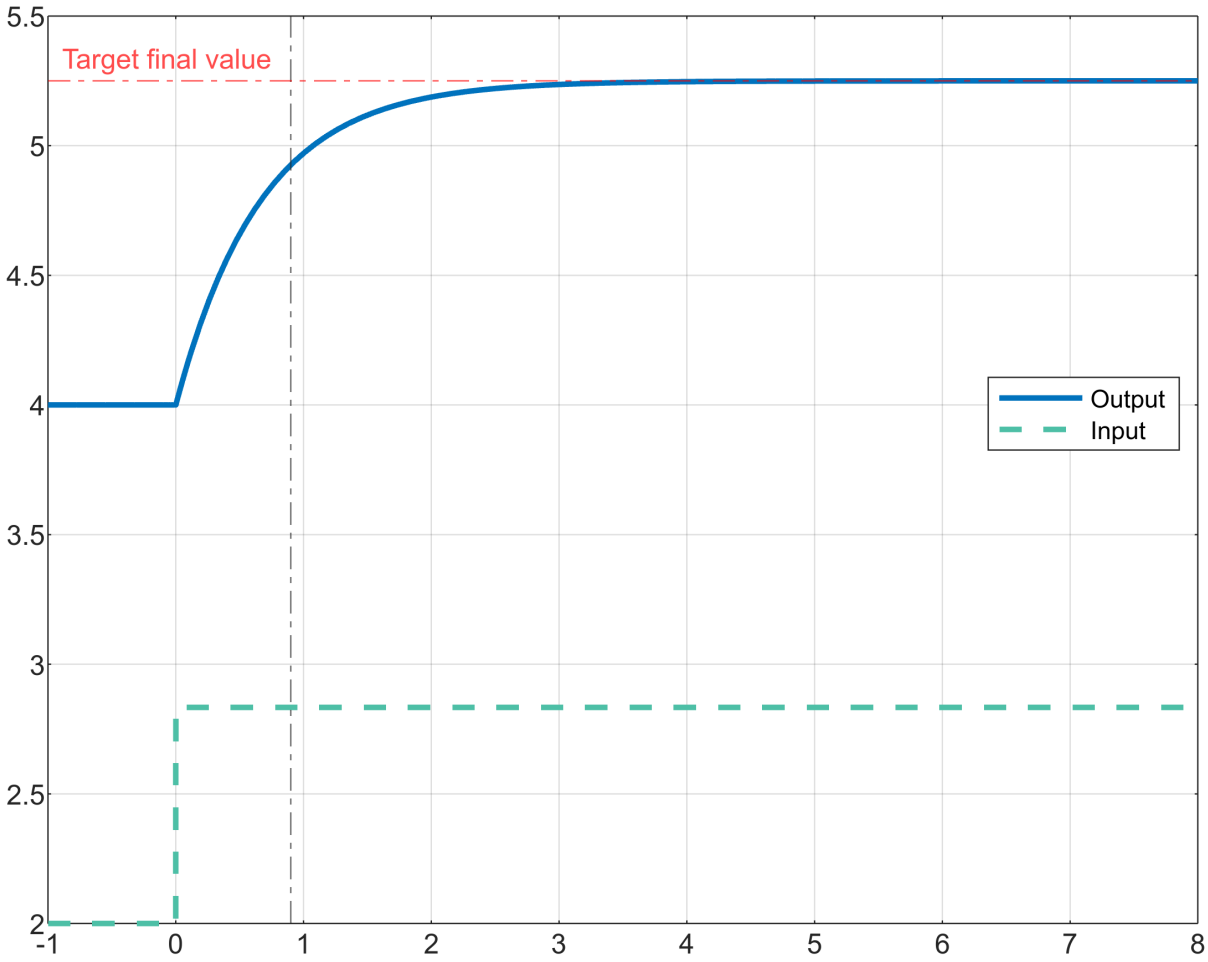
```
computed_inc_input = 0.8333
```

Therefore, the "absolute" (i.e., non-incremental) input value that in equilibrium will achieve the desired output will be:

```
u_computed=u_op+computed_inc_input
```

```
u_computed = 2.8333
```

```
u=@(t) u_op+computed_inc_input*(t>=0);
simulsystem(u);
xline(desired_settling_time,'-.')
yline(5.25,'-.r',Label="Target final value",LabelHorizontalAlignment="left")
legend("Output","Input",Location="best")
```



Preset final value and settling time

1.) What input will be needed to raise the output to **5.25** units in **0.9 seconds**?

By "proportionality" arising from linear systems, if with an increment of 0.5 it goes up 0.55 units in 0.9 seconds, to go up to 5.25 in the same time, we need:

```
inc_output=4.555-y_op %experimental data
```

```
inc_output = 0.5550
```

```
gain_in_1dot4seconds=inc_output/inc_u %increment per unit input in given time
```

```
gain_in_1dot4seconds = 1.1100
```

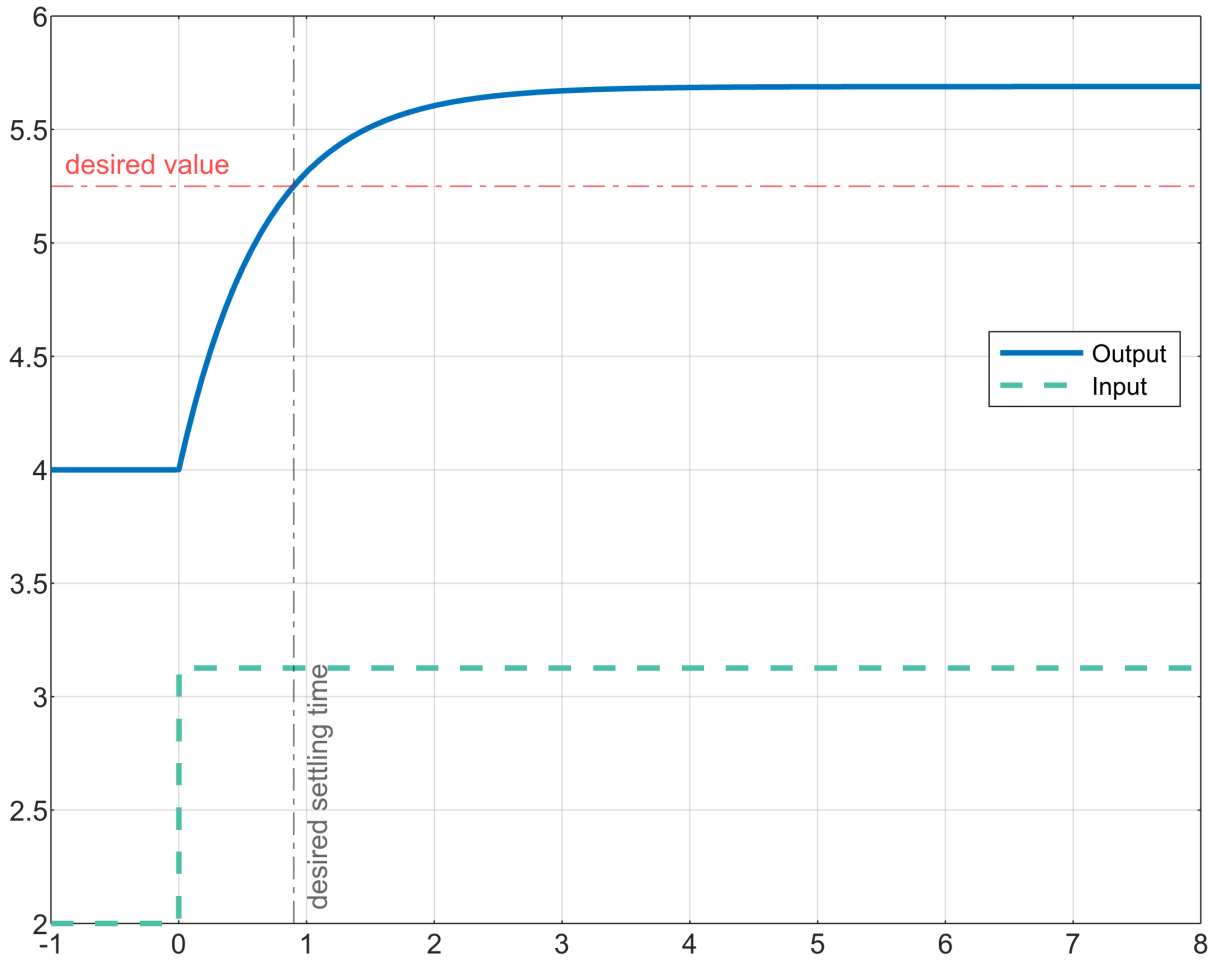
```
desired_inc_output=5.25-y_op
```

```
desired_inc_output = 1.2500
```

```
inc_input_computed2=desired_inc_output/gain_in_1dot4seconds
```

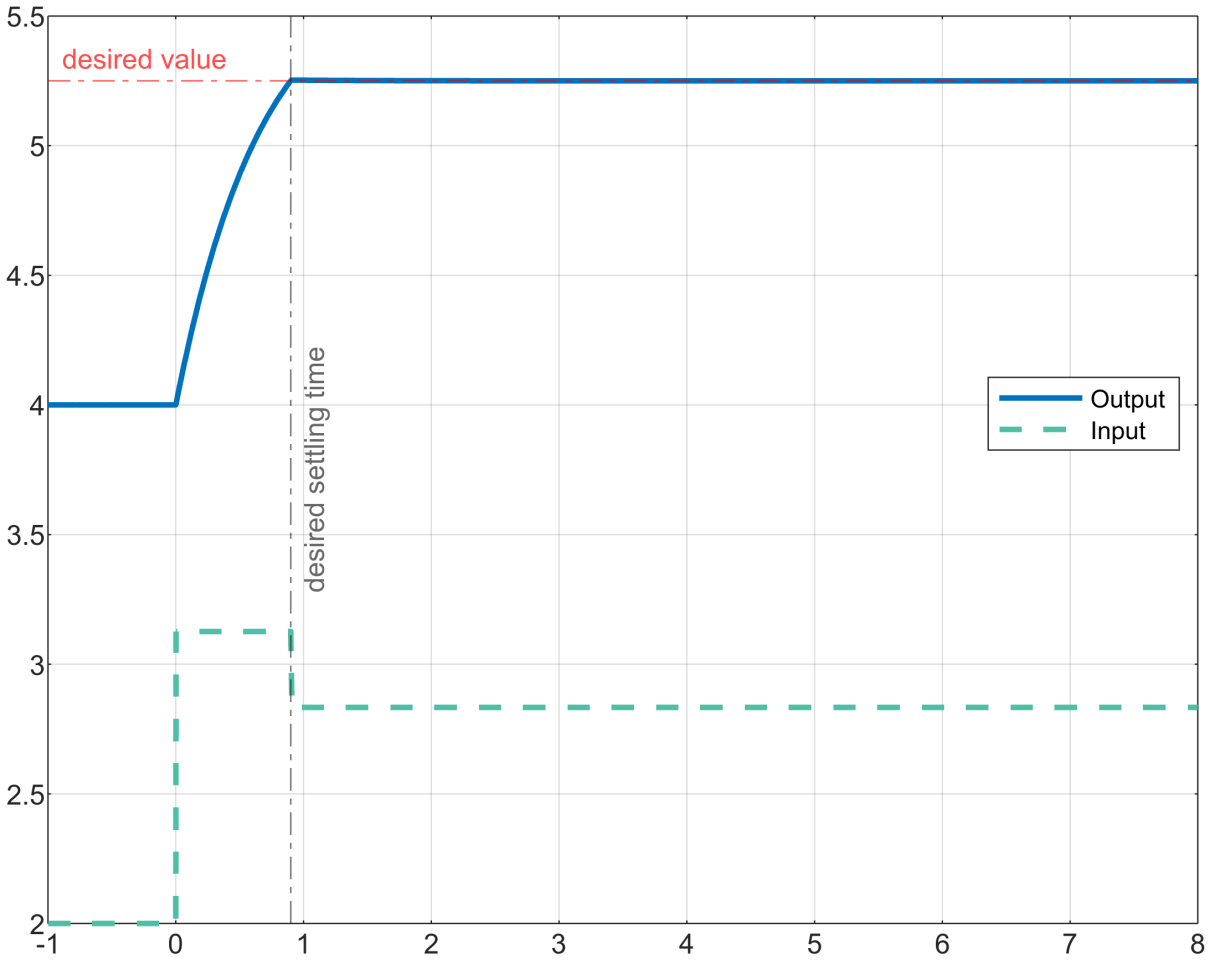
```
inc_input_computed2 = 1.1261
```

```
u=@(t) u_op+inc_input_computed2*(t>=0);
simulsystem(u);
yline(5.25,'-.r',Label="desired value",LabelHorizontalAlignment="left")
xline(desired_settling_time,'-.',Label="desired settling time",LabelVerticalAlignment="
legend("Output","Input",Location="best")
```



In order to avoid exceeding the desired final value, it will be necessary to switch to the input value that maintains the desired value in equilibrium once it is reached:

```
u=@(t) u_op+inc_input_computed2*(t>=0).*(t<=desired_settling_time)+computed_inc_input*
simulsystem(u);
yline(5.25,'-.r',Label="desired value",LabelHorizontalAlignment="left")
xline(desired_settling_time,'-.',Label="desired settling time",LabelVerticalAlignment="
legend("Output","Input",Location="best")
```



That is, we have designed a 2-step profile: an initial "boost" to go up faster and then a "permanent" value to stay at the desired point. This is a "precomputed" input profile (open-loop control): no measurement is taken while the step sequence is being applied in order to decide when to switch.

NOTE: the computations we made are only "exact" in first-order linear systems; in higher order linear systems there is a certain "inertia" that will mean that even if the input is lowered to the calculated equilibrium point, there will be a certain "transient overshoot". This will be illustrated in a companion video linregla3ord2EN.html .

Appendix: auxiliary functions

This code is, supposedly, "*secret*": it's not needed to examine it in order to carry out the computations intended to be the goal of this material. This is a sort of abstraction of doing an "experiment":

```
function dxdt=model1(x,u)
 dxdt=-1.5*x+2.25*u+1.5;
end
```

```
function Y=simulsystem(u)
opts=odeset(Reltol=1e-5,AbsTol=1e-5);
[T,X]=ode45(@(t,x) model1(x,u(t)),[-1 8],4,opts);
plot(T,X,LineWidth=2), grid on
Y=X;
hold on
plot(T,u(T'),'--',LineWidth=2,Color=[.3 .75 .65])
hold off
end
```