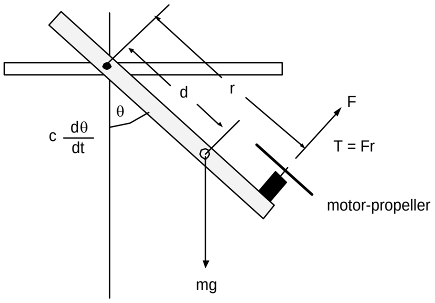
Hands-on Lab:

LabVIEW – Simulink PID

Proportional-Integral-Derivative (PID) control is the most used method to make a system perform as desired. In a previous lab, Simulink was introduced to simulate the LEGO Damped Compound Pendulum (DCP). That lab concluded: (1) the theoretical simulated response matched those of the DCP's experimental results; (2) the DCP's step response showed the pendulum oscillating but eventually reached a steady-state angle, but took a long time. This lab applies and simulates PID control of the DCP.

Preamble: PID and DCP Block Diagram

For reference, as illustrated in the previous lab, Figure A, B, and C:



r	Pendulum Length	0.265 [m]
d	Pivot to CG distance	0.1325 [m]
m_L	Mass of Pendulum	0.0137 [kg]
ζ	Damping ratio	0.033
ω_n	Natural frequency	7.32 [rad/s]
J	Moment of Inertia	0.00033 [kgm^2]
c	Coefficient of Friction	0.00016 [Nms/rad]
θ_{ss}	Steady-state angle	31^o at 50% motor power

Figure A: DCP free body diagram (left) and both physically measured and calculated values (right)

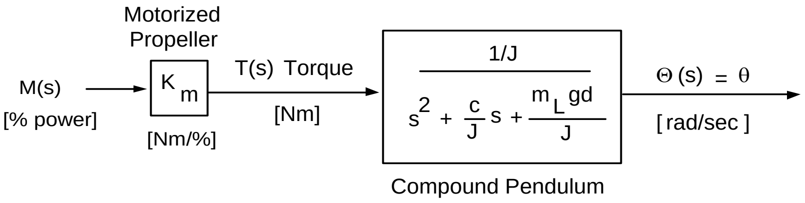


Figure B: Block diagram relating the motor power input $\text{M}(s)$, to the DCP's angle output $\Theta(s)$

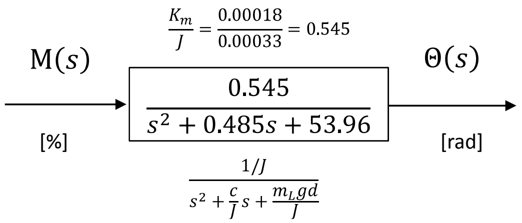

$\frac{K_m}{J} = \frac{0.00018}{0.00033} = 0.545$
$\text{M}(s)$
[%]
$\frac{0.545}{s^2 + 0.485s + 53.96}$
$\Theta(s)$
[rad]
$\frac{1/J}{s^2 + \frac{c}{J}s + \frac{m_L gd}{J}}$

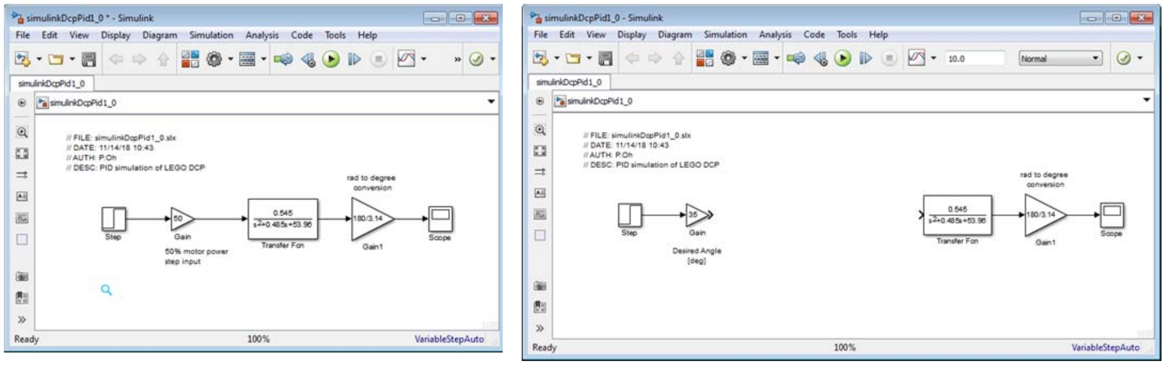
Figure C: Transfer function relating the motor input (as a percentage) and angle output (in radians)

The box in **Figure C** is called the open-loop transfer function (OLTF) and relates the input (motor power command in units of %) to the output (DCP's angle in units of radians). The specific values in the box's numerator and denominator were specific to the DCP used in the previous lab. These values may differ for another DCP, depending the mass, pendulum length, and motor-prop properties. As such, one would replace the values with obtained from system identification, into **Figure B** to yield **Figure C**.

Concept 1: Simulink Block Diagram

Step 1: Open Simulink and Create the Open-loop Transfer Function (OLTF)

The previous lab created `simulinkDcpStepInput1_0.slx` (**Figure 1-1A left**) which had the OLTF. Save this Simulink file as `simulinkDcpPid1_0.slx` and delete the wire between the 50% gain block and the OLTF (**Figure 1-1A right**). For this simulation, let us set the desired DCP angle for 35^0. Thus, double-click the 50% gain, and under `Block Parameters`, replace 50 with 35. Edit the text box, to underscore that this gain represents a desired DCP angle of 35^0.

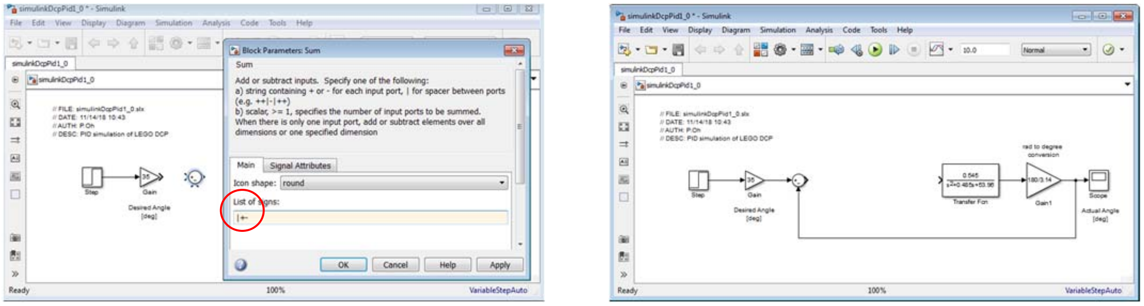


// FILE: simulinkDcpPid1_0.slx
// DATE: 11/14/18 10:43
// AUTH: P.Oh
// DESC: PID simulation of LEGO DCP

Figure 1-1A: The former Step input Simulink program `simulinkDcpStepInput1_0.slx` already created the OLTF (left). Rename the 50% gain block, edit comments and save as `simulinkDcpPid1_0.slx` (right).

Step 2: Add the Error Signal

PID control operates by performing functions on the error signal. This signal is the difference between the desired and actual angles. Under the `Library Browser`, select `Commonly Used Blocks` and click-and-drag the `Sum` block into the Simulink canvas. By default, the `Sum` block adds 2 inputs. Double-click the `Sum` block and in `Block Parameters` change the last addition to a subtraction (**Figure 1-2A left**) and wire blocks (**Figure 1-2A right**).



Block Parameters: Sum

Sum

Add or subtract inputs. Specify one of the following:
a) string containing + or - for each input port, | for spacer between ports (e.g. ++|-|++)
b) scalar, >= 1, specifies the number of input ports to be summed.
When there is only one input port, add or subtract elements over all dimensions or one specified dimension

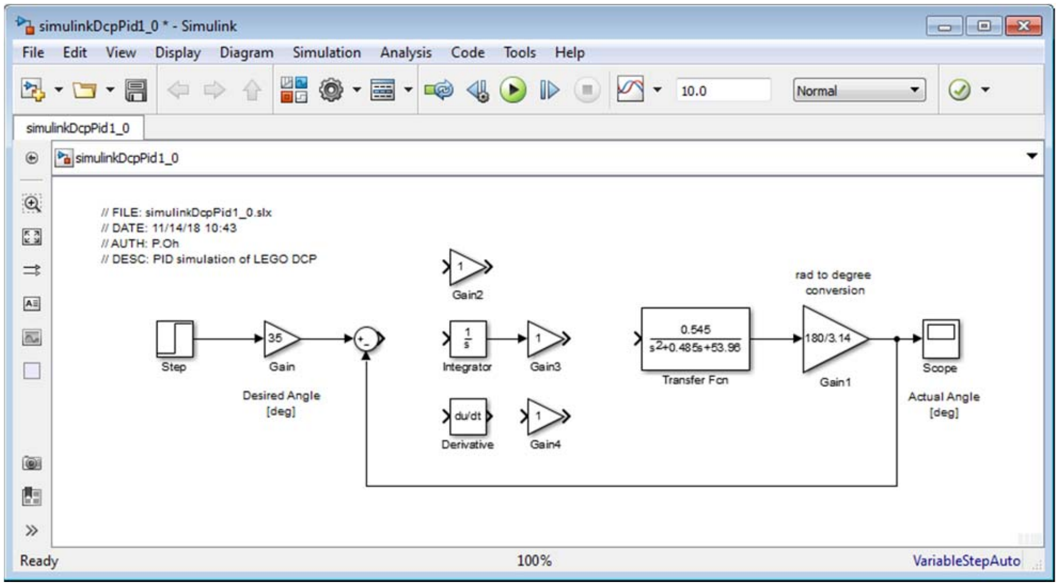
Icon shape: round

List of signs: |+-

Figure 1-1A: Red circle shows where to configure the Sum block to subtract signals (left). Wiring (right) now reflects an error signal that is the difference between Desired and Actual angles [deg]

Step 3: Create the Proportional, Integral and Derivative Blocks

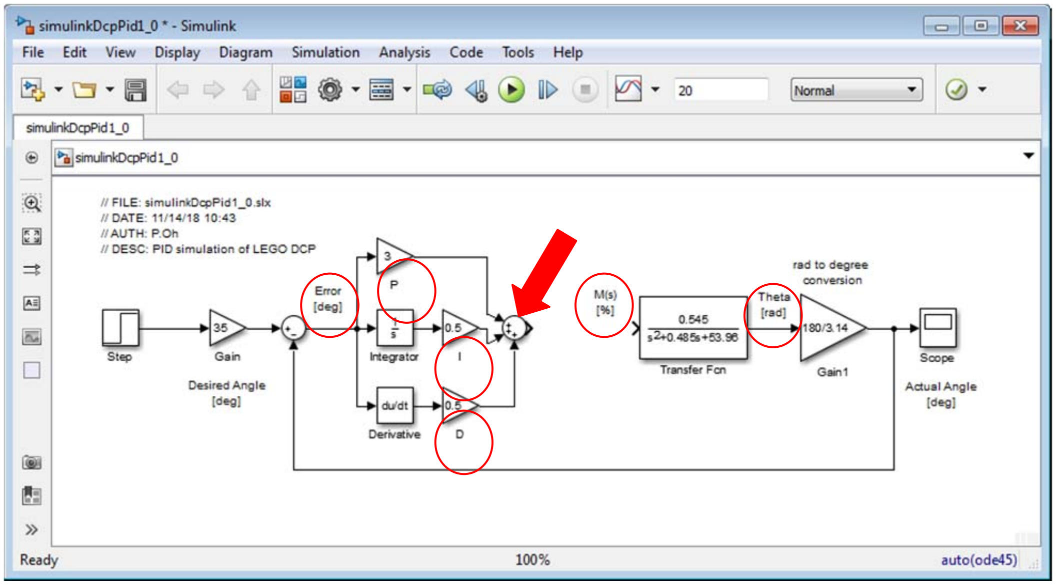
PID respectively multiplies, integrates and differentiates the error signal. Under the `Library Browser – Continuous`, click-and-drag the `Derivative` and `Integrator` blocks into the Simulink canvas. Under `Commonly Used Blocks`, click-and-drag the three `Gain` blocks. Rearrange blocks to resemble **Figure 1-3A**.



```
// FILE: simulinkDcpPid1_0.slx
// DATE: 11/14/18 10:43
// AUTH: P.Oh
// DESC: PID simulation of LEGO DCP
```

Figure 1-3A: Proportional simply multiples the error with a constant, hence a Gain block is used. Integral performs a scaled integration on the error, hence the Integrator and Gain blocks. Lastly, Derivative performs scaled differentiation on the error, hence the Derivative and Gain blocks.

Next, add a second `Sum` block. Double-click it and under `Block Parameters`, and a + so that the `Sum` block allows the addition of 3 signals. Add text comments for blocks and wires to identify signals and units (**Figure 1-3B**).

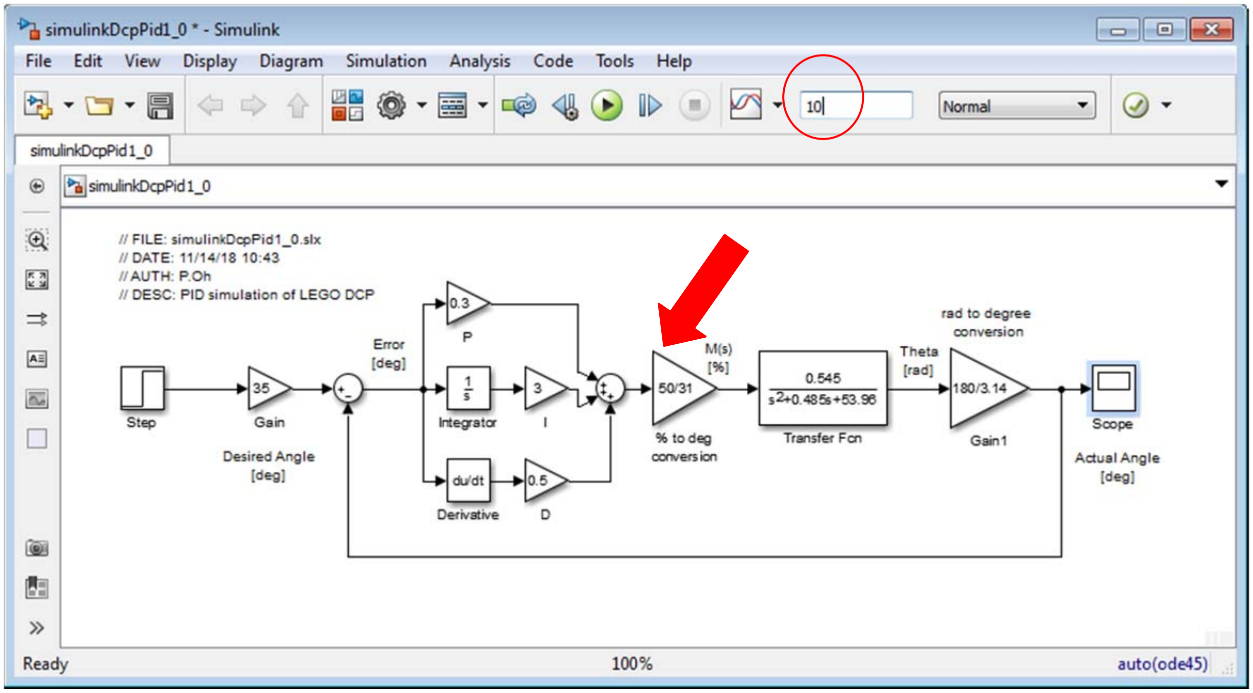


```
// FILE: simulinkDcpPid1_0.slx
// DATE: 11/14/18 10:43
// AUTH: P.Oh
// DESC: PID simulation of LEGO DCP
```

Figure 1-3B: Add and configure a second Sum Block so that it can add the 3 signals from the P, I, and D blocks (red arrow). Add text comments to clarify each signal and units (e.g. M(s) and [%]) in the appropriate areas (red circles)

As **Figure 1-3B** shows, the output of the PID summation block will be in degrees. However, the input $M(s)$ into the OLTF is a percentage (i.e. motor power). Recall in previous derivations, a 50% motor power input yielded a steady-state angle of 31 degrees. Assuming linearity, these values can be used to convert degrees to percentage, namely $\frac{50\%}{31\text{deg}} = 1.61\ \%/\text{deg}$.

Add a `Gain` block between the PID summation and the OLTF and configure it be `50/31` (**Figure 1-3C**). Save the resulting program (i.e. `simulinkDcpPid1_0.slx`).



```
// FILE: simulinkDcpPid1_0.slx
// DATE: 11/14/18 10:43
// AUTH: P.Oh
// DESC: PID simulation of LEGO DCP
```

Figure 1-3C: Red arrow points to the Gain block used to convert the PID control signal (in degrees) to a motor power input $M(s)$ in percentage. Simulation time (red circle) was set to 10 seconds.

Step 4: Execute PID

Figure 1-3C also shows gains values to scale the proportional, integration, and differentiation. These values are respectively $[P, I, D = [03, 3.0, 0.5]]$. The figure also sets the simulation to 10 seconds. Saving and executing the program results in some overshoot and convergence to the desired steady-state angle $\theta_{ss} = \theta_{\text{desired}} = 35^0$ (**Figure 1-4A**).

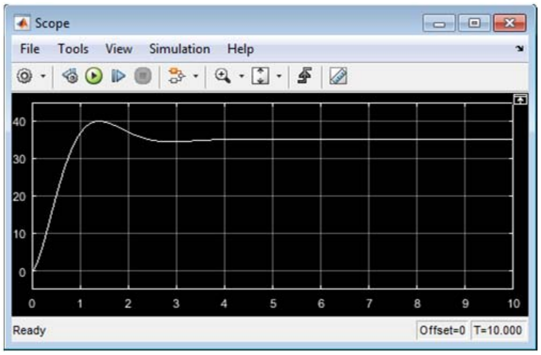


Figure 1-4A: PID output for $[P, I, D = [03, 3.0, 0.5]]$. One sees that after about 3 seconds, $\theta_{ss} = \theta_{\text{desired}} = 35^0$

Congratulations: LEGO DCP PID Simulation Completed

Exercise 1:

1.1 What PID gain values will result in no overshoot i.e. $\theta_{\text{actual}} \leq \theta_{\text{desired}}$. Show your resulting figure

1.2 Add an additional scope to display the value of $M(s)$in percentage for the gain values you chose in 1.1. If this $M(s) > 100\%$re-do 1.1 with PID gains such that the motor power command remains between 0 and 100%

1.3 What PID gain values will result in a reduced settling time i.e. $t_s < 2$ seconds. Show your resulting figure.

1.4 If this $M(s) > 100\%$ re-do 1.3 with PID gains such that the motor power command remains between 0 and 100%