**Hands-on Lab:**

**LabVIEW – Simulink Testing**

Previously, one captured both the step and free fall response of the LEGO Damped Compound Pendulum (DCP). Using the logarithmic decrement equation and the DCP’s physical measurements, one calculated the damping ratio, coefficient of friction, and moment of inertia (**Figure 1-1a**). Such system identification can now be used to determine the DCP (i.e. plant) transfer function. Equipped with this transfer function, one can use simulation tools, like Simulink, for future controller design.

**Concept 1:** **Calculating the DCP transfer function**

**Figure 1-1a**: DCP free body diagram (left) and both physically measured and calculated values (right)



|  |  |  |
| --- | --- | --- |
| $$r$$ | 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 |

A second-order system is characterized by the equation (1):



(1)

From the free body diagram, the motor-prop generates a torque$ T$. Thus dynamic equilibrium is given by (2):



(2)

Linearizing (2) where $\sin(θ≈θ) $for small$ θ$, and matching coefficients in equations (1) and (2) reveals the following relationships

$$c=2ζω\_{n}J$$

(3a)

$$J=\frac{m\_{L}gd}{ω\_{n}^{2}}$$

(3b)

Taking the Laplace transform of (2) yields

Hence

$$s^{2}Θ\left(s\right)+s\frac{c}{J}Θ\left(s\right)+\frac{m\_{L}gd}{J}Θ\left(s\right)=Τ\left(s\right) or \left(s^{2}+\frac{c}{J}s+\frac{m\_{L}gd}{J}\right)Θ\left(s\right)=Τ\left(s\right)$$

$$\frac{Τ\left(s\right)}{Θ\left(s\right)}=\frac{^{1}/\_{J}}{Js^{2}+cs+m\_{L}gd}$$

(4)

In block diagram form, (4) looks like **Figure 1-1b**. The torque $ Τ\left(s\right)$ results from the thrust force applied to the DCP’s lever arm $r $(i.e. the pendulum’s length). This thrust is a result of a voltage applied to the motor-prop and prop size. The NXT Brick creates this voltage using a power command (in units of %) ranging from 0 to 100. Let $Μ\left(s\right)$ represent this motor power.

**Figure 1-1b:** Block diagram relating the motor power input$ Μ\left(s\right)$, to the DCP’s angle output $ Θ\left(s\right)$



One could analytically use the prop’s dimensions (e.g. diameter, number of blades, and pitch) to calculate lift. If can assume$ M\left(s\right) $and$ Τ\left(s\right) $are linearly related and use a proportionality constant (motor gain) $ K\_{m} $for the relationship as in (5)

(5)

$$K\_{m}=\frac{Τ\left(s\right)}{Μ\left(s\right)}$$

To determine $K\_{m} $one can recognize that at dynamic equilibrium, when the DCP is at its steady-state angle$ θ\_{ss}$, (2) becomes:

$$m\_{L}gd\sin(θ\_{ss}=T|\_{ss})$$

(6)

In the previous lab, as shown in Figure 1-1a (last row of table), for a $Μ\left(s\right)=50\% $step input, yielded a $θ\_{ss}=31^{o}$. Substituting this into (6) and (5) yields

$$K\_{m}=\frac{m\_{l}gd\sin(31^{o})}{50\%}=\frac{0.0137kg∙9.81\frac{m}{s^{2}}∙0.1325m∙0.515}{50\%}=0.00917 Nm$$

(7)

Using the values from (7) and the table in **Figure 1-1a**, the resulting block diagram is given in **Figure 1-1c.**

**Figure 1-1c**: Transfer function relating the motor input (as a percentage) and angle output (in radians)



**Exercise 1:**

* 1. Redo the table in Figure 1-1a, using the values you observed and calculated
	2. Using 1.1, calculate the motor gain $K\_{m}$
	3. Sketch Figure 1-1c, using and answers to 1.1 and 1.2

**Concept 2:** Simulink Testing

Simulink is Matlab’s graphical programming environment. XCOS is the equivalent, for Scilab, a free open-source version. Simulink and XCOS while not exactly the same, have very similar blocks. For this concept, Simulink[[1]](#footnote-1) will be used.

**Step 1:** Launch Matlab and Open Simulink

In Matlab R2016, launching the program will reveal the opening screen (**Figure 2-1A left**) and clicking the Simulink icon brings up the Simulink Start Page (**Figure 2-1A right**).

**Figure 2-1A:** Matlab launch allows one to click on the Simulink icon, indicated by the red arrow (left). This launches the Simulink Start Page (right)





Clicking on New Model launches a blank canvas screen (**Figure 2-1B left**). From the canvas’ menu bar, click View – Library Browser. A menu of blocks is then revealed (**Figure 2-1B right**)

**Figure 2-1B:** Red arrow points to the menu bar’s View selection (left). Clicking Library Browser reveals the blocks one can use in a Simulink simulation (right)





Before populating the Simulink canvas, save the file as simulinkDcpStepInput1\_0.slx.

**Step 2:** Populate the Simulink Canvas – Transfer Function

Using the Library Browser, click Continuous, select and drag the Transfer Fcn block into the canvas (**Figure 2-2A left**). Double-clicking any white space in the canvas, allows one to add text. Commenting your program with details is a good practice. Double-clicking the Transfer Fcn block pops-up the Block Parameter box. Referring to **Figure 1-1C** and your answers from **Exercise 1-3**, enter the Numerator and Denominator coefficients (**Figure 2-2A right**).

**Figure 2-2**: Transfer Fcn block and text comments in the Simulink canvas (left). Double-clicking the Transfer Fcn block reveals the Block Parameters pop up box (right). The numerator and denominator coefficients were entered using derived values shown in **Figure 1-1C**.





Make the Transfer Fcn block slightly bigger, by clicking and dragging a corner. One will then be able to the see the transfer function values in the block.

**Step 3:** Populate the Simulink Canvas – Gains

Next, under Commonly Used Blocks, click and drag 2 gains. Double-click each gain to specify their values. Also connect the gain and Transfer Fcn blocks by pointing the mouse to an output and dragging the resulting wire into an input (**Figure 2-3A**).

Figure 2-3A: Gain blocks before and after the Transfer Fcn block, with input and outputs wired



**Step 4:** Add Step Input and Output Scope

In the Library Browser, click Sources and click-and-drag the Step block into the Simulink canvas and wire it to the gain. Double-click the Step block and set Step time to 0. Click Sinks and similarly bring a Scope into the Simulink canvas and wire it to the gain. Save the resulting program (**Figure 2-4A**) as simulinkDcpStepInput1\_0.slx.

**Figure 2-4A:** Finished Simulink program. Setting the Step time to 0, sends a step input value of 1, which is multiplied by the gain 50. This represents a 50% motor power input into the DCP transfer function. Since the output of this transfer function is in radians, a rad-to-degree gain is used. The result is then displayed on a scope.



**Step 5:** Execute Simulink program

Hitting the play button (red arrow in Figure 2-4A) will execute the program and when completed, a chime will sound. Double-clicking the scope will display the output (**Figure 2-5A**).

**Figure 2-5A**: Simulink output (left) shows steady-state$ θ\_{ss}≈31^{o}$. In the previous lab, an experimental step response was captured and plotted (right). The two show similarities, given confidence in the transfer function derived in **Figure 1-1C**.





**Congratulations: LEGO Damped Compound Pendulum Simulation Completed**

**Exercise 2:**

* 1. What are some of the reasons that may explain the differences in Figure 2-5A?
	2. Study Matlab/Simulink to change the scope’s black background to white and add labels to the axes. Hint 1: Right-click on the figure, and select Style. Hint 2: In Matlab, type
	3. Study Matlab and try to overlap the CSV data and Simulink scope output
1. Matlab R2016a version was used for these notes [↑](#footnote-ref-1)