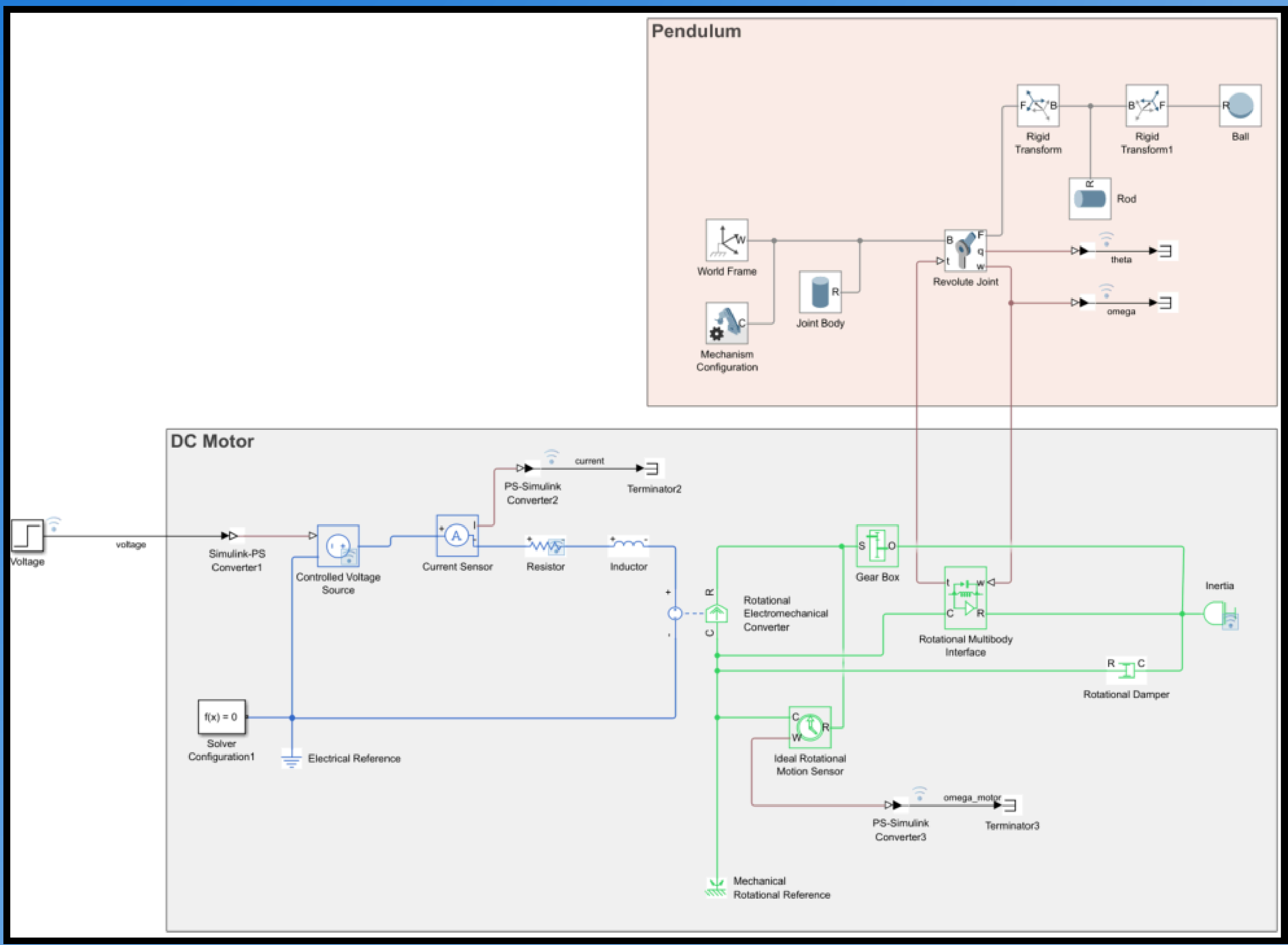
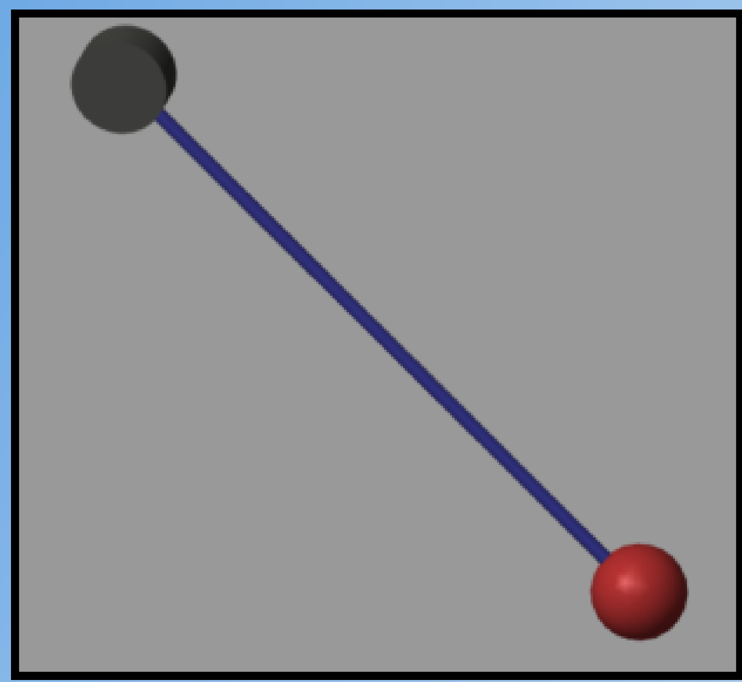
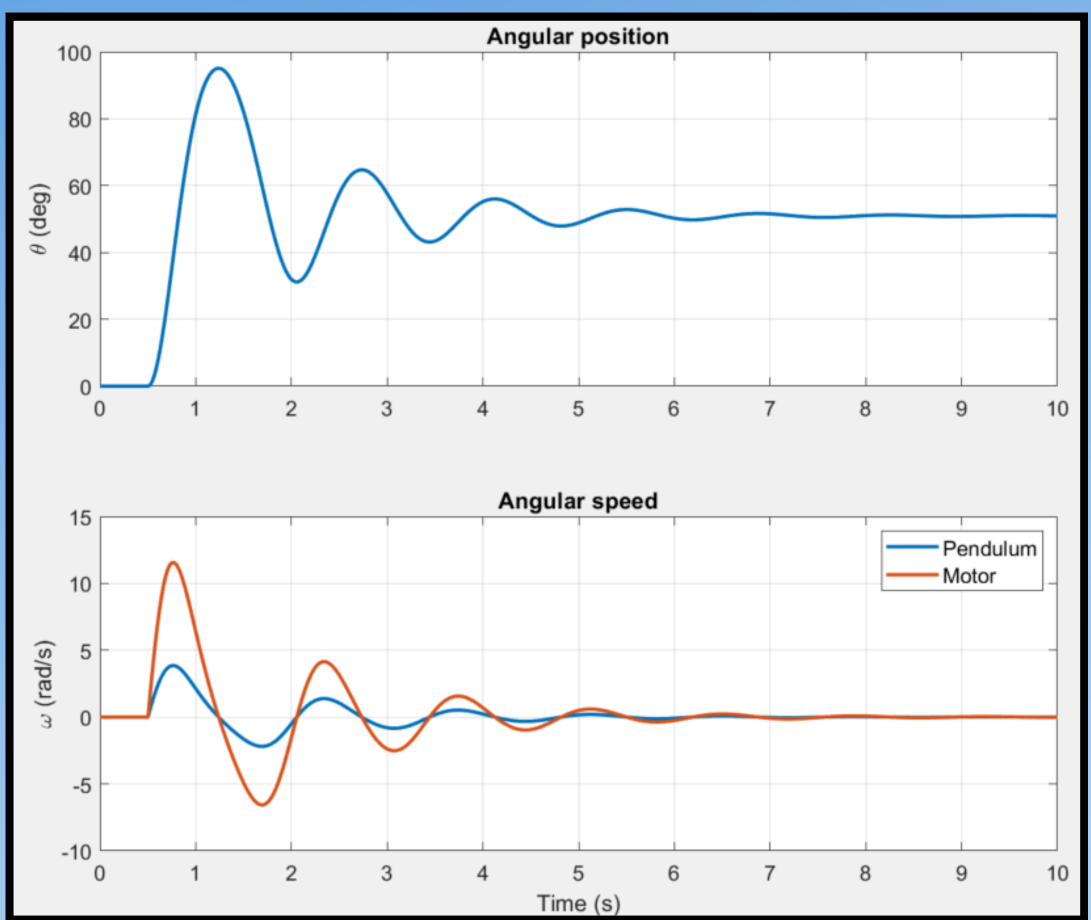
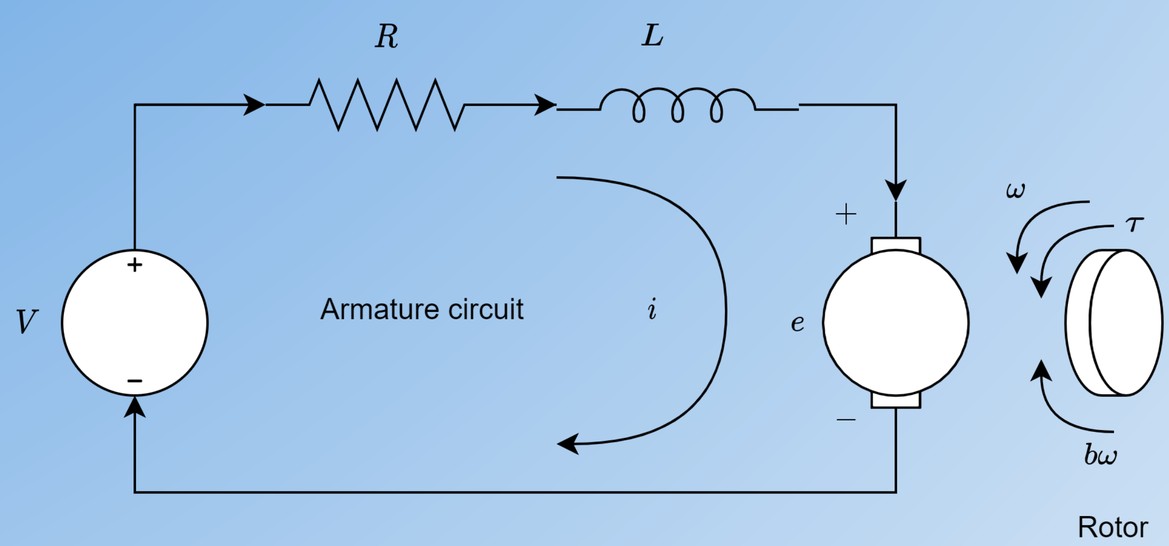
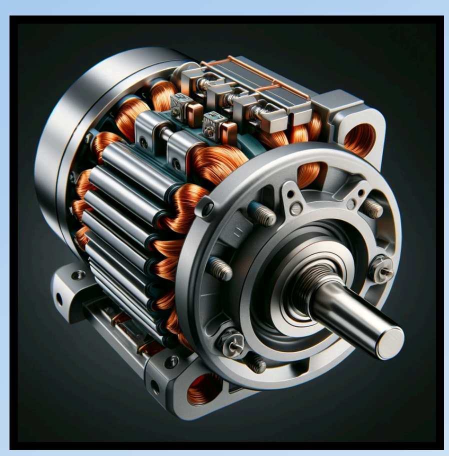
Pendulum – DC Motor Simscape









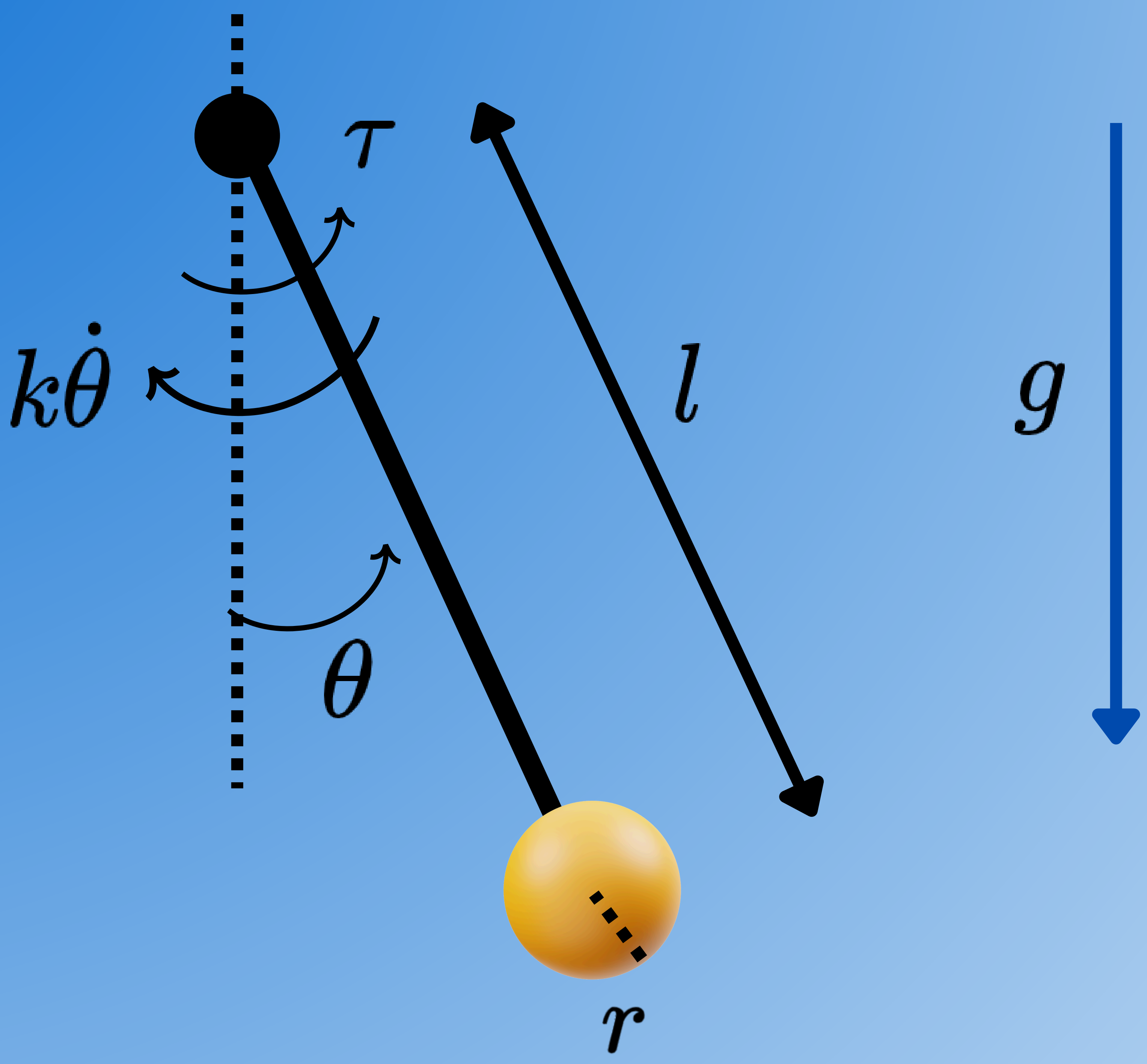


Model

https://github.com/simorxb/pendulum-dc-motor-simscape

SIMONE BERTONI CONTROL LAB

Pendulum



Ball radius -> $r = 0.02\ m$

Rod length -> $l = 0.3\ m$

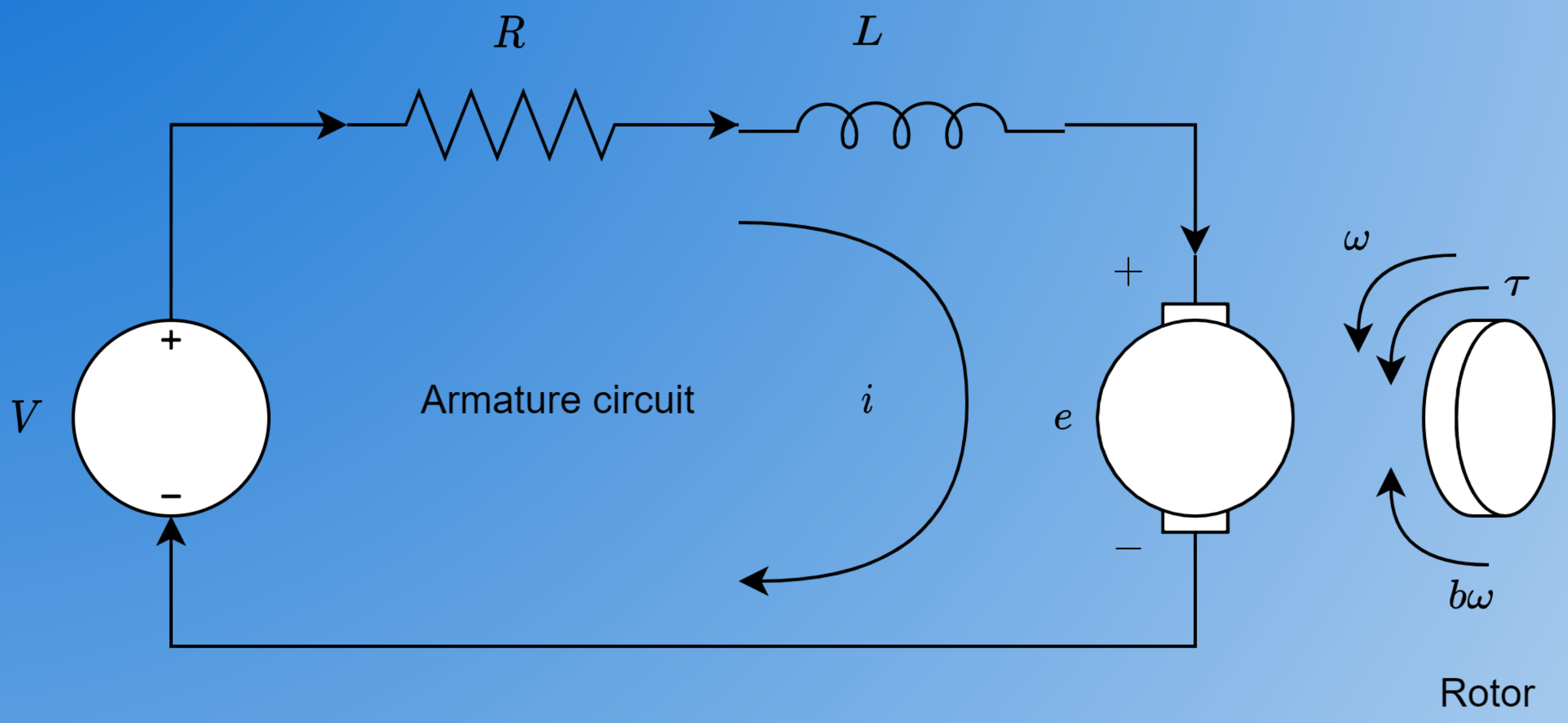
Rod diameter -> $d = 0.003\ m$

Joint damping coefficient -> $k = 0.03\ Nms$

Density (Steel) -> $\rho = 7750\ \frac{kg}{m^3}$

Gravity -> $g = 9.81 \frac{m}{s^2}$

DC Motor – Model



Mechanical - Newton's second law for rotational motion:

$$J\dot{\omega} + b\omega = \tau, \; \tau = k_t i$$

Electrical:

$$L\frac{di}{dt} + Ri = V - k_e\omega$$

Isolate the highest level derivatives to facilitate modelling:

$$\dot{\omega} = \frac{k_t i - b\omega}{J}$$

$$\frac{di}{dt} = \frac{V - k_e\omega - Ri}{L}$$

Include angular position:

$$\dot{\theta} = \omega$$

DC Motor – Parameters

Referring to the datasheet of a real DC motor (C23-L33-W10) from Moog (https://www.moog.com/content/dam/moog/literature/MCG/moc23series.pdf) we can derive our parameters:

Torque sensitivity (k_t) = 0.0187 Nm/A

Back EMF (k_e) = 0.0191 V/(rad/s)

Terminal resistance (R) = 0.6 Ohm

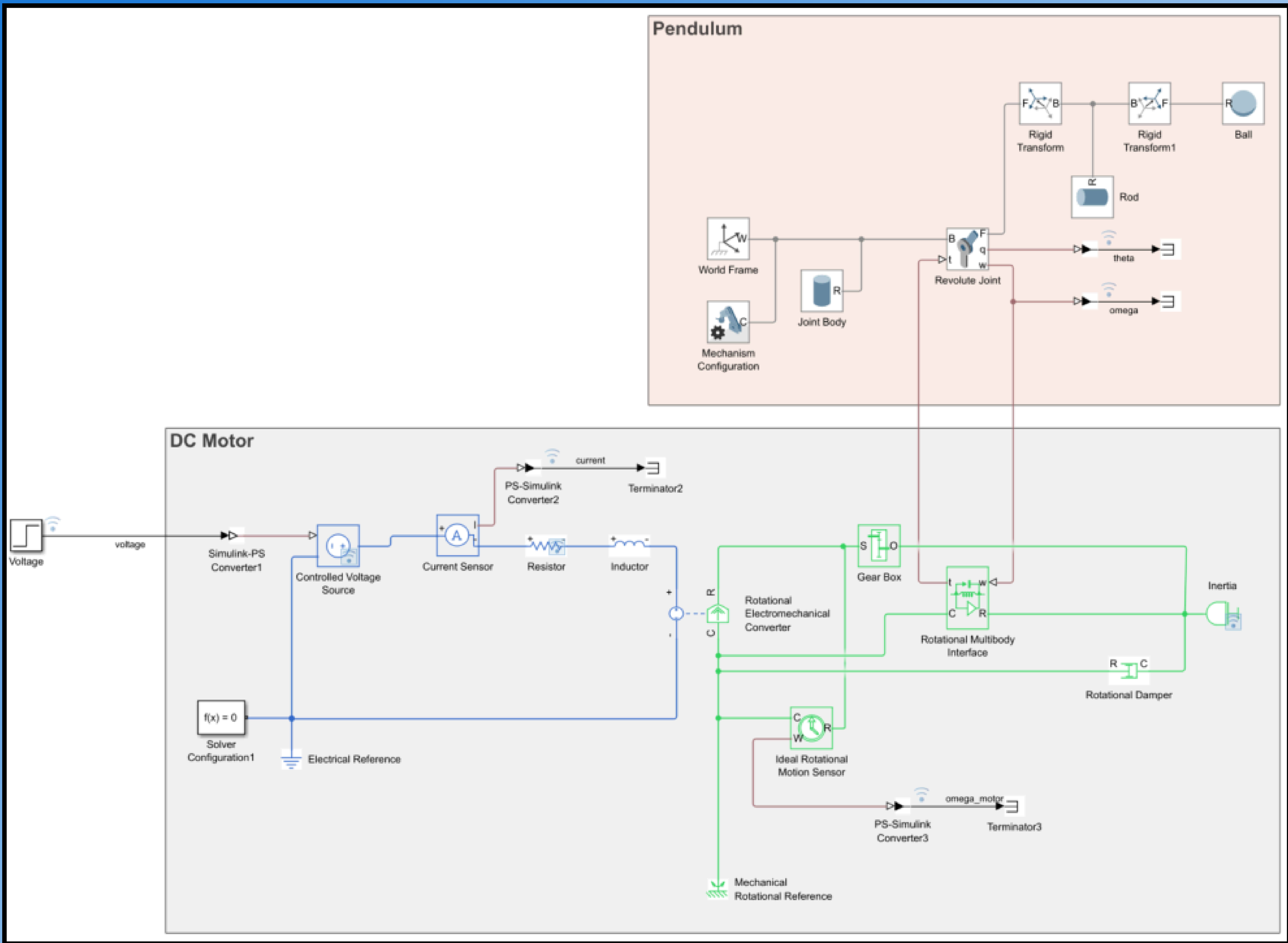
Terminal inductance (L) = 35 mH = 0.035 H

Damping factor (b) = 0.001 Nm/KRPM = 0.0000095 Nm/(rad/s)

Assuming that we are spinning a disc of radius 5 cm and mass 0.1 kg, we have:

$$J = 0.5mr^2 = 0.000125\ kgm^2$$

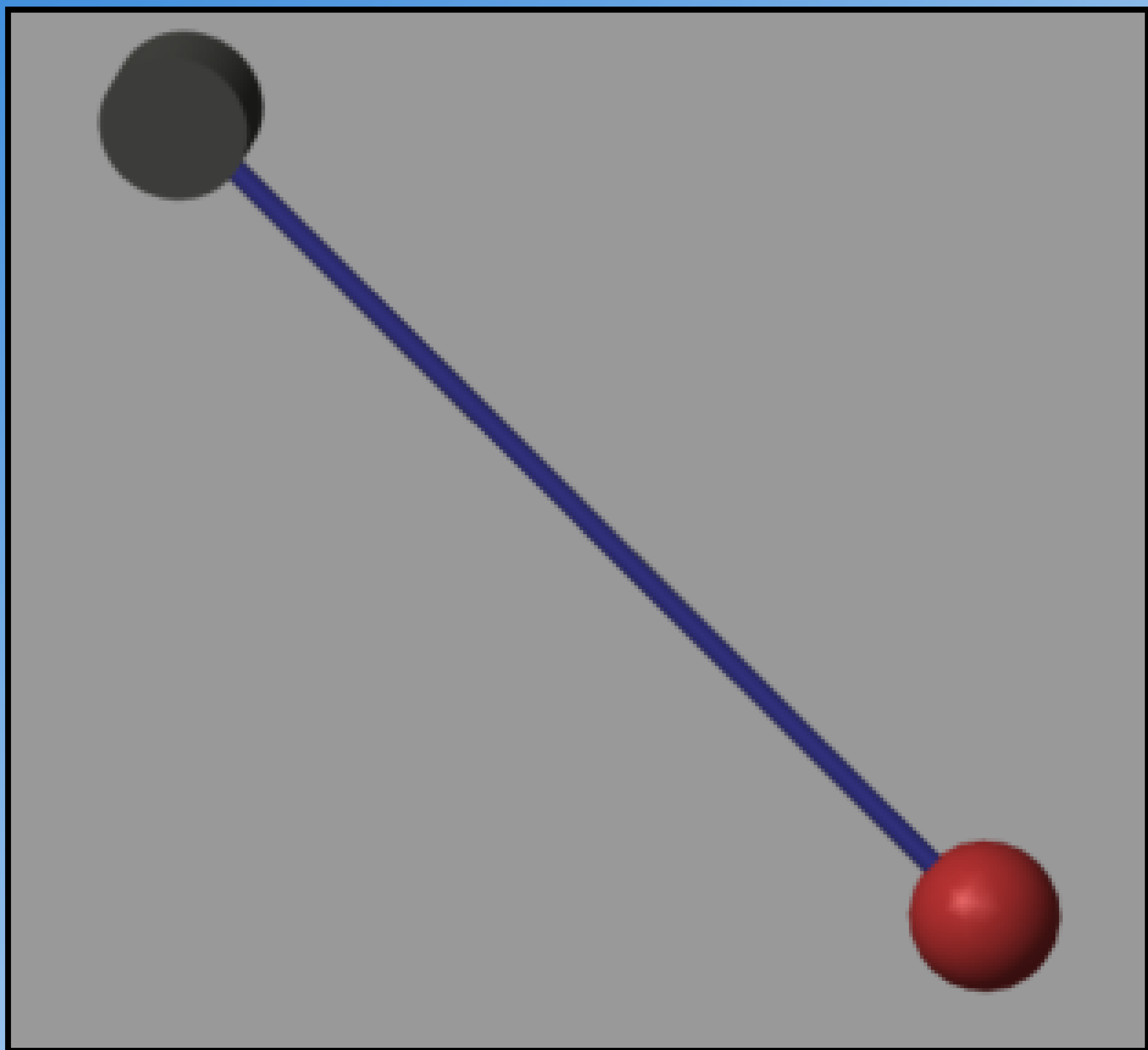
Simscape Model



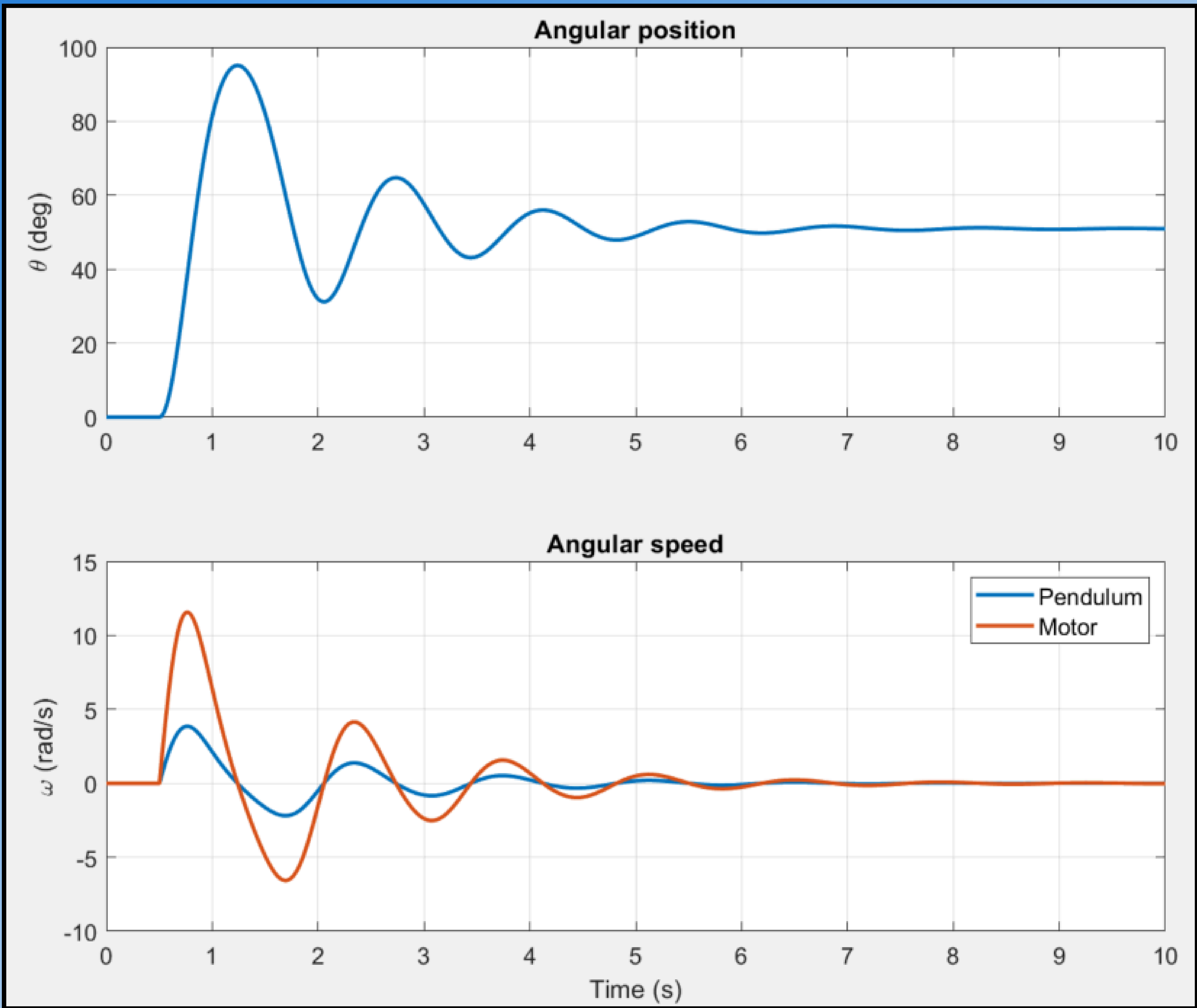
New elements

Rotational Multibody Interface: to connect DC motor torque/speed to multibody revolute joint.
Gear Box: to “amplify” DC Motor torque.

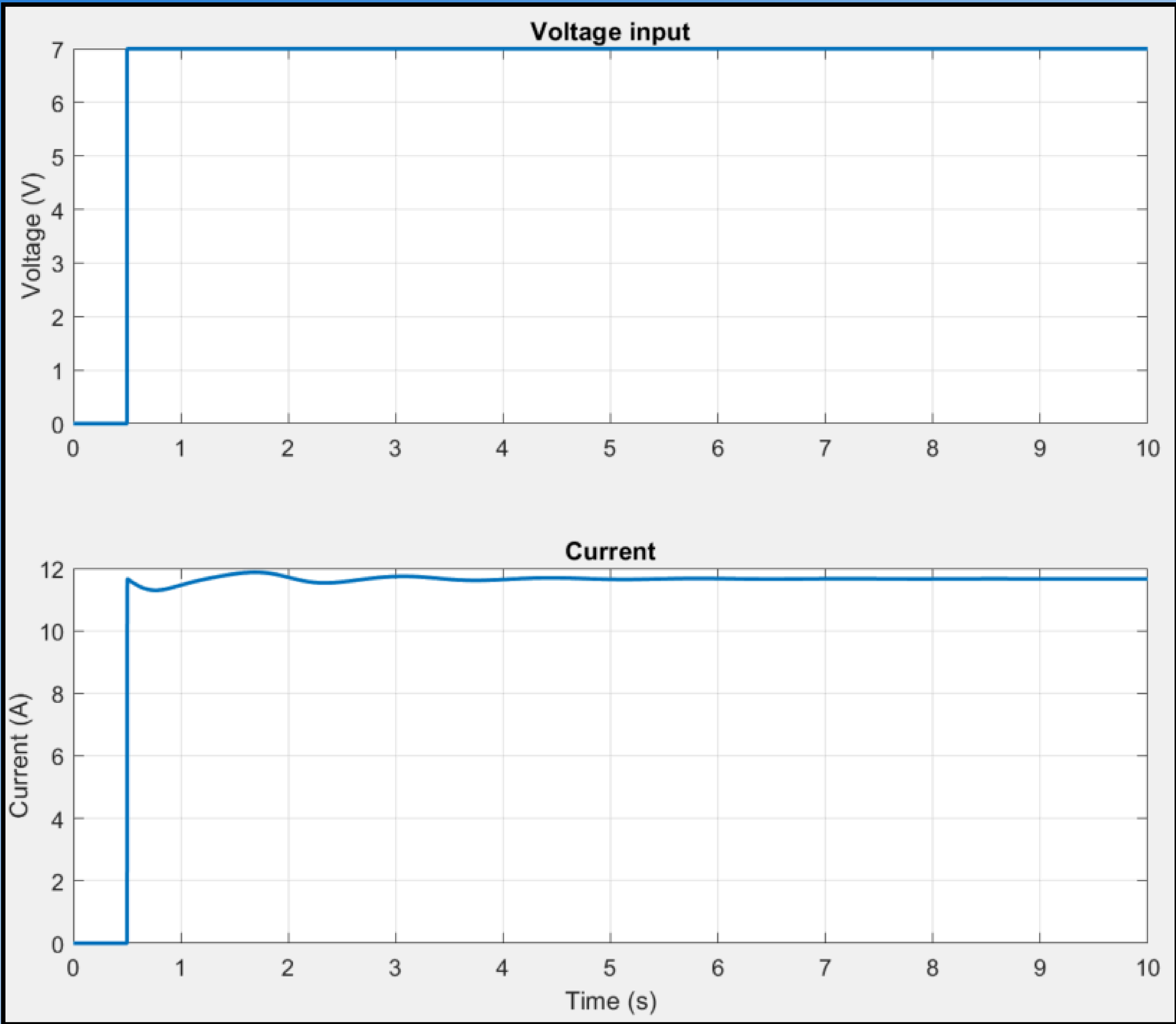
Simscape Multibody Mechanics Explorer



Simulation Result



Simulation Result



Matlab Code Initialization

```matlab
%% Init params - Pendulum

% Joint damping coefficient
k = 0.03;

% Rod length
l = 0.3;

% Rod diameter
d = 0.003;

% Ball radius
r = 0.02;

% Rod and ball density (kg/m^3) - Steel
rho = 7750;

%% Init params - DC Motor

% Motor inertia
J = 2.2e-10;

%Motor damping coefficient
b = 0.0000095;

% Torque sensitivity
kt = 0.0187;

% Resistance
R = 0.6;

% Inductance
L = 0.35/1000;

% Back EMF
ke = 0.0191;
```

Matlab Code
Plot Results

```matlab
% Access the signals from out.logsout

theta = out.logsout.get('theta').Values.Data;
t_theta = out.logsout.get('theta').Values.Time;
omega = out.logsout.get('omega').Values.Data;
t_omega = out.logsout.get('omega').Values.Time;
omega_motor = out.logsout.get('omega_motor').Values.Data;
t_omega_motor = out.logsout.get('omega_motor').Values.Time;
voltage = out.logsout.get('voltage').Values.Data;
t_voltage = out.logsout.get('voltage').Values.Time;
current = out.logsout.get('current').Values.Data;
t_current = out.logsout.get('current').Values.Time;

% Create the first figure
figure;

% Subplot for theta
subplot(2, 1, 1);
plot(t_theta, theta, 'LineWidth', 2);
ylabel('\theta (deg)');
set(gca, 'FontSize', 12);
title('Angular position');
grid on;

% Subplot for omega
subplot(2, 1, 2);
plot(t_omega, omega, 'LineWidth', 2);
hold on;
plot(t_omega_motor, omega_motor, 'LineWidth', 2);
hold off;
ylabel('\omega (rad/s)');
xlabel('Time (s)');
legend({'Pendulum', 'Motor'}, 'FontSize', 12);
set(gca, 'FontSize', 12);
title('Angular speed');
grid on;

% Create the second figure
figure;

% Subplot for voltage
subplot(2, 1, 1);
plot(t_voltage, voltage, 'LineWidth', 2);
ylabel('Voltage (V)');
set(gca, 'FontSize', 12);
title('Voltage input');
grid on;

% Subplot for current
subplot(2, 1, 2);
plot(t_current, current, 'LineWidth', 2);
hold off;
ylabel('Current (A)');
xlabel('Time (s)');
title('Current');
set(gca, 'FontSize', 12);
grid on;
```

PID Controller Course

https://simonebertonilab.com

Understand the control theory

★★★★★ *April 28, 2024*

I think the most important thing is to understand the meaning behind the mathematical formula. I guess this is the mission of Simone in this course and from my point of view he fully achivied this target. I hope to see in the future other courses (e.g advanced controls) structered in the same way with the same passion and examples.

Thank you Simone. Show less

Emidio Verified

★★★★★

Very helpful and practical

Yoav Golan

I enjoyed this course very much. I learned a lot of practical knowledge in a short time. Simone is very clear and teaches well, thank you! In the future, I would be very interested if Simone added a course with more subjects, such as cascading controllers, rate limiting, and how the controllers look in actual code. Thanks again!

★★★★★

Intuitive and Practical

Ranya Badawi

Simone's explanation of PID control was very intuitive. This is a great starter course to gain a fundamental understanding and some practical knowledge of PID controllers. I highly recommend it. For future topics, I'd be interested in frequency response, transfer functions, Bode plots (including phase/gain margin), Nyquist plots, and stability.

★★★★★

Very good sharing of experience

Romy Domingo Bompart Ballache

I have background in control system for power electronics, I see every lesson very useful.

Great course

★★★★★ *April 15, 2024*

Right to the point, easy to follow and very practical. I missed the zero/pole placement and phase margin analysis. It would also be interesting if you could provide other plants examples. Anyway, a great course to help designing and tuning a PID controller.

Leonardo Starling Verified

A different way to learn PID !

★★★★★ *May 31, 2023*

The teacher explains PID in a clear way adding his experience there where formulas alone cannot do much. Furthermore, each topic covered is included in a practical example to better fix ideas.

Michele De Palma