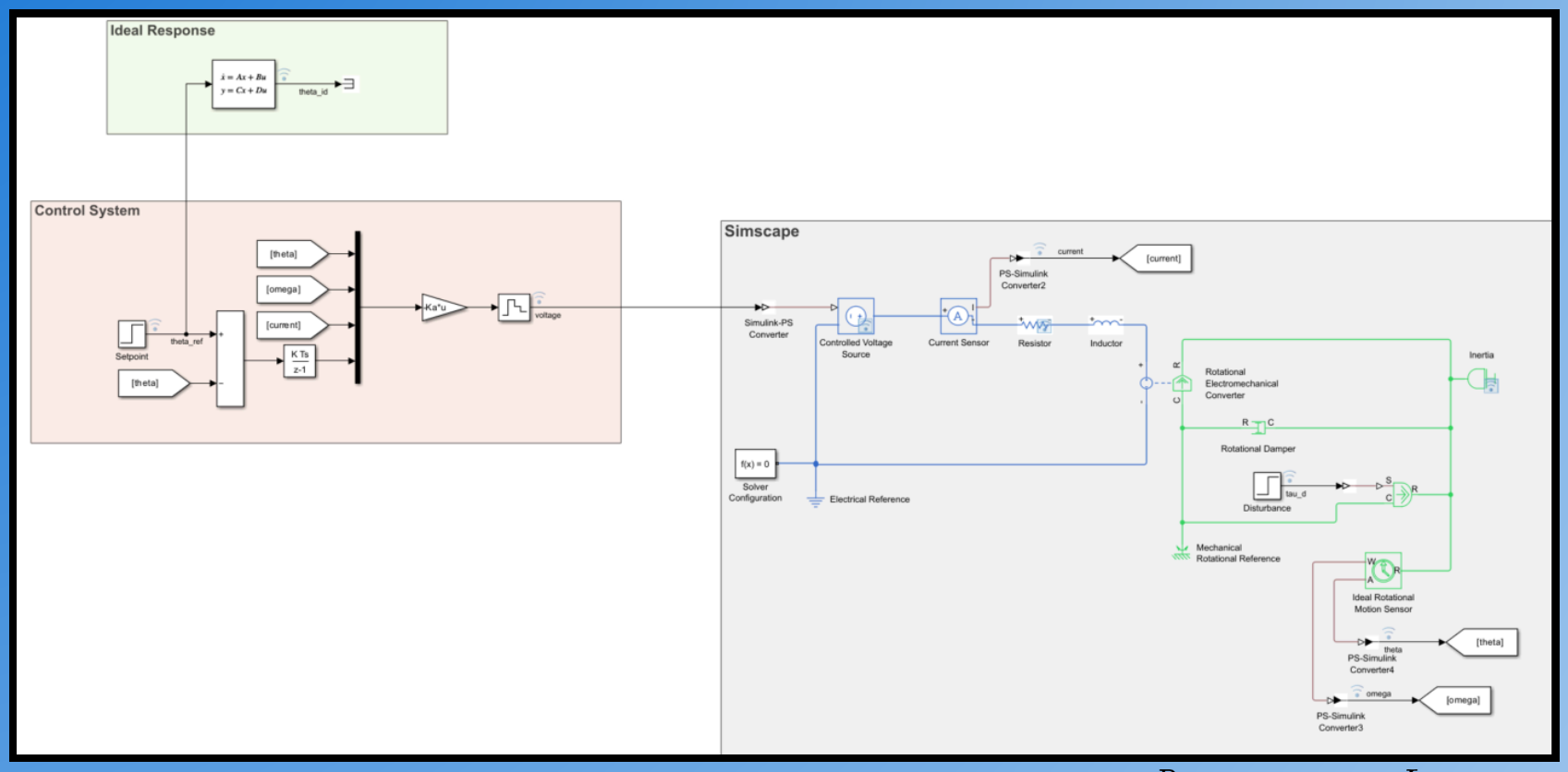
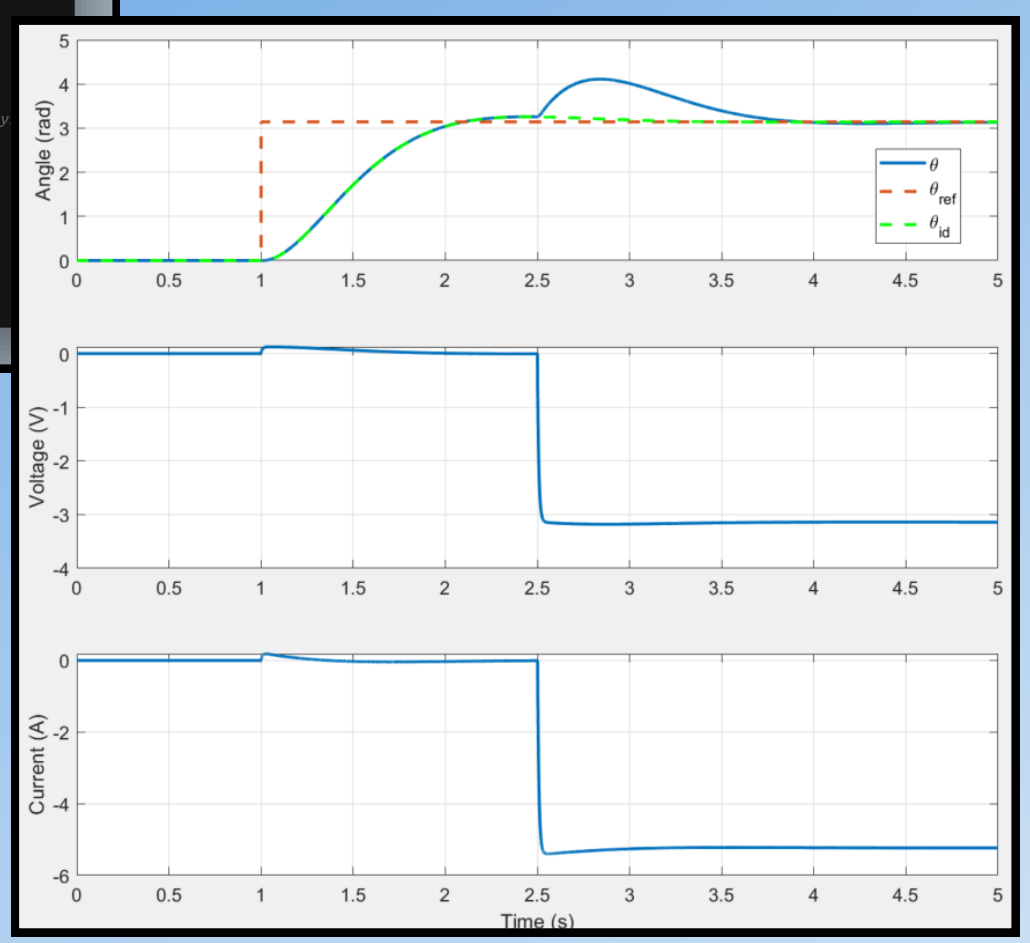
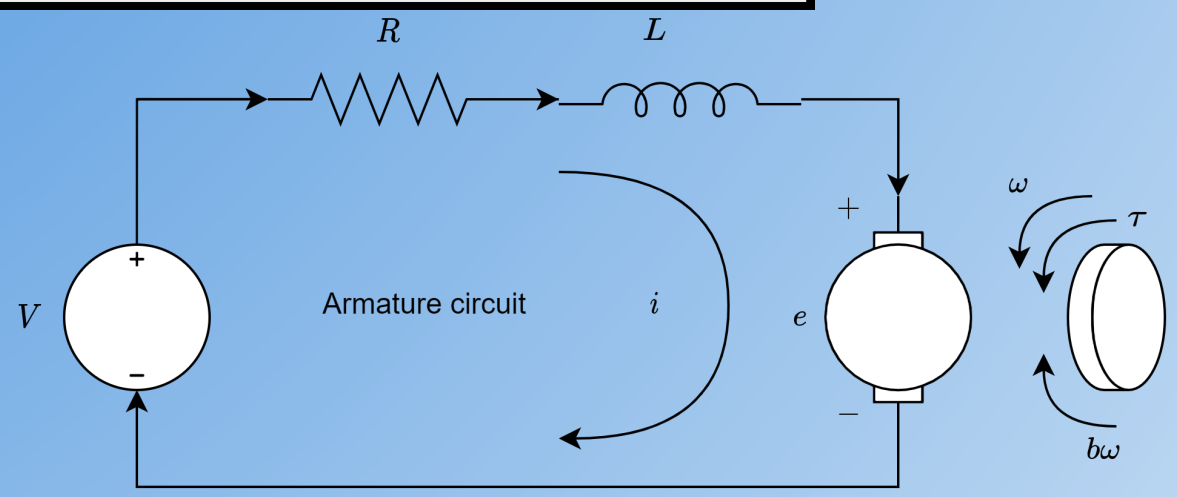
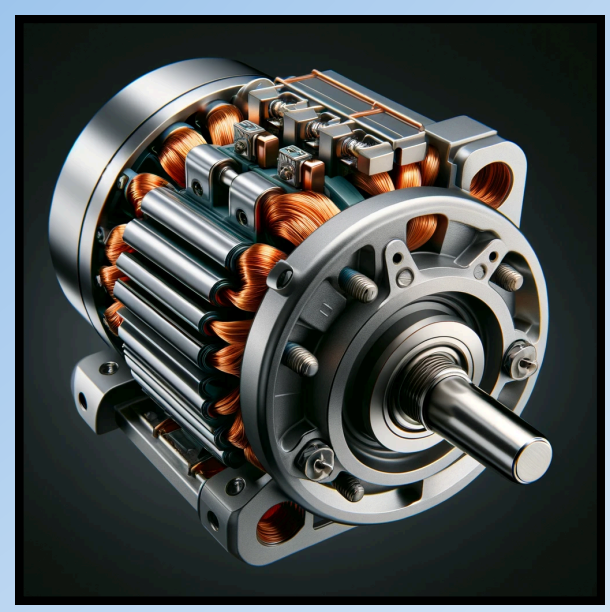
DC Motor
LQR - Offset-Free




```m
%% Init params
m = 0.1;
r = 0.05;
J = 0.5*m*r^2;
b = 0.0000095;
kt = 0.0187;
R = 0.6;
L = 0.35/1000;
ke = 0.0191;

%% Controller params

% Compute A, B, C, D matrices
A = [0 1 0; 0 -b/J kt/J; 0 -ke/L -R/L];
B = [0; 0; 1/L];
C = [1 0 0];
D = 0;

% Compute A, B, C, D matrices for augmented system
Aa = [A, zeros(3,1); -C, 0];
Ba = [B; -D];
Ca = [C 0];

% Compute state feedback gains using LQR for augmented sy
Q = diag([1 1 1 100]);
R_ = 1;
Ka = lqr(Aa, Ba, Q, R_);

% Calculate ideal response subsystem
idResp = ss(Aa - Ba*Ka, [zeros(3,1); 1], Ca, D);

tSamp = 0.001;
```





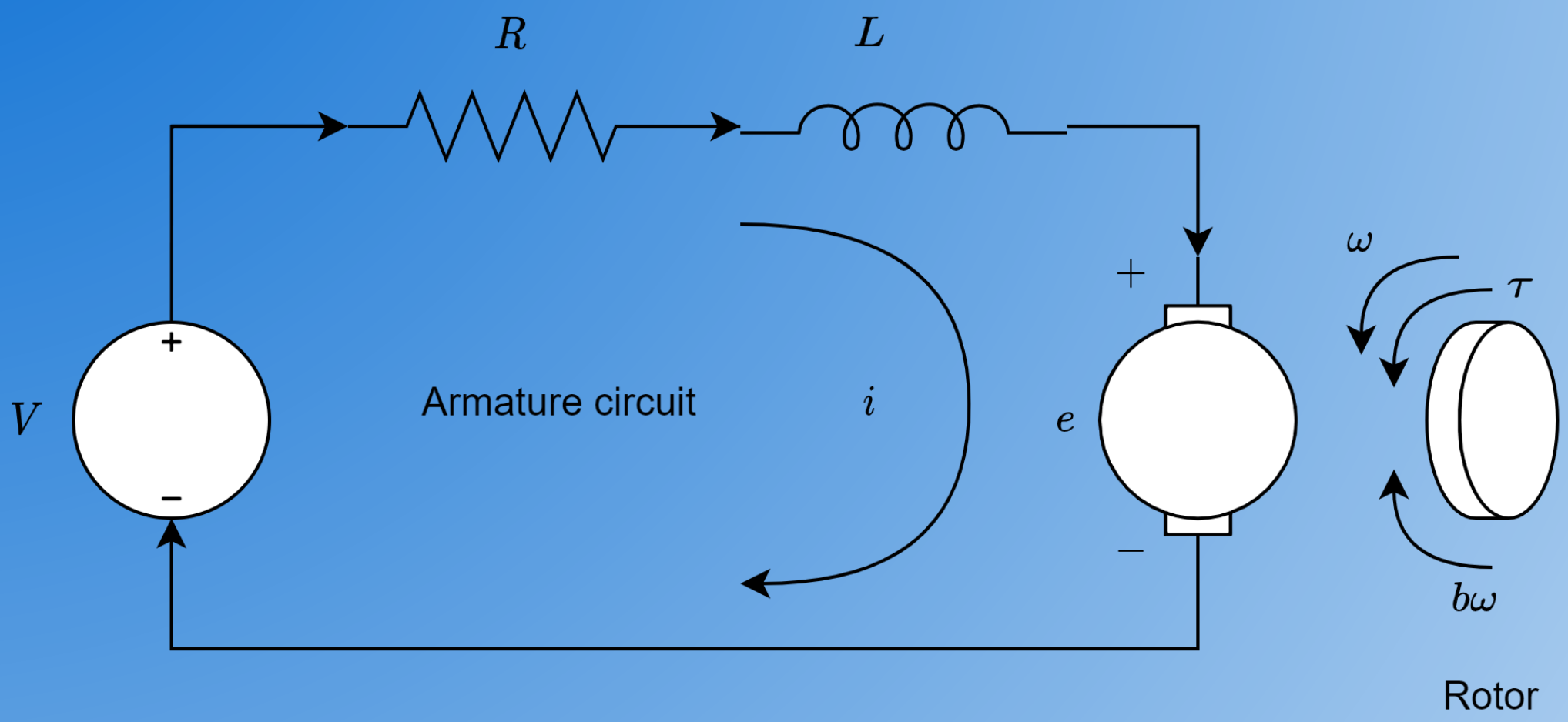


Model

https://github.com/simorxb/dc-motor-simscape-lqr-offset-free

SIMONE BERTONI
CONTROL LAB

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:

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

State Space Model

Linear dynamic system - state space model:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}$$

$$\mathbf{y} = \mathbf{C}\mathbf{x} + \mathbf{D}\mathbf{u}$$

DC motor as a state space model:

$$\mathbf{x} = \begin{bmatrix} \theta \\ \omega \\ i \end{bmatrix}$$

$$\mathbf{u} = V$$

$$\mathbf{y} = \theta = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} x$$

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & -\frac{b}{J} & \frac{k_t}{J} \\ 0 & -\frac{k_e}{L} & -\frac{R}{L} \end{bmatrix}$$

$$\mathbf{B} = \begin{bmatrix} 0 \\ 0 \\ \frac{1}{L} \end{bmatrix}$$

$$\mathbf{C} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}$$

$$\mathbf{D} = 0$$

State Feedback Control

State feedback control:

$$\mathbf{u} = \mathbf{K_r} r - \mathbf{K} x$$

where r is the setpoint. Now the equivalent system, found by substitution, is:

$$\dot{\mathbf{x}} = (\mathbf{A} - \mathbf{BK})\mathbf{x} + \mathbf{BK_r r}$$

$$\mathbf{y} = (\mathbf{C} - \mathbf{DK})\mathbf{x} + \mathbf{DK_r r}$$

Using this approach, we can choose $\mathbf{K}$ to assign the poles of the "new" system. This can be done in Matlab using "place", for example:

K = place(A, B, [-10 -11 -12]);

Now we need to choose $\mathbf{K_r}$ to have $G(0) = 1$ where:

$$G(s) = \frac{Y(s)}{R(s)} = (C + DK)(sI - A + BK)^{-1}BK_r + DK_r$$

and we find:

$$\mathbf{K_r} = \frac{1}{(C+DK)(-A+BK)^{-1}B+D}$$

State Feedback Control + Integral Action – 1

The approach shown above has a weakness, steady steate control error in case of constant disturbance or modelling error (i.e. always).

That's because there is no integral action in the control law.

To improve this control technique we can add an integral part to the control scheme, adding a new state x_{n+1} (the integral of the control error) to the system with an integral gain K_{n+1}:

$$\dot{x}_{n+1} = r - (Cx + Du).$$

Now we select a control law u as

$$u = -Kx - K_{n+1}x_{n+1},$$

where K is a $1 \times n$ vector of constant gains. Substituting in $\dot{x}_{n+1}$ gives:

$$\dot{x}_{n+1} = r - (C - DK)x + DK_{n+1}x_{n+1}$$

State Feedback Control + Integral Action – 2

Now, substituting into the open-loop state equation, we obtain:

$$\dot{x} = Ax - B(Kx + K_{n+1}x_{n+1})$$

Rearranging results in

$$\dot{x} = (A - BK)x - BK_{n+1}x_{n+1}$$

And the augmented state-space model can be written as

$$\begin{bmatrix} \dot{x} \\ \dot{x}_{n+1} \end{bmatrix} = \begin{bmatrix} A - BK & -BK_{n+1} \\ -C + DK & DK_{n+1} \end{bmatrix} \begin{bmatrix} x \\ x_{n+1} \end{bmatrix} + \begin{bmatrix} \Theta \\ 1 \end{bmatrix} r$$

where Θ is an $n \times 1$ vector of zeros. Hence, the closed-loop dynamics become

$$\begin{cases} \dot{x}_a = (A_a - B_a K_a)x_a + \begin{bmatrix} \Theta \\ 1 \end{bmatrix} r \\ y_a = C_a x_a \end{cases}$$

State Feedback Control + Integral Action – 3

The matrices A_a, B_a, C_a, and K_a are given by:

$$A_a = \begin{bmatrix} A & \Theta \\ -C & 0 \end{bmatrix},$$

$$B_a = \begin{bmatrix} B \\ -D \end{bmatrix},$$

$$C_a = \begin{bmatrix} C - DK & -DK_{n+1} \end{bmatrix},$$

$$K_a = \begin{bmatrix} K & K_{n+1} \end{bmatrix}.$$

Linear Quadratic Regulator LQR

A Linear Quadratic Regulator (LQR) problem is a fundamental concept in control engineering and optimization. It is used to design a feedback controller that minimizes a quadratic cost function of the state and control variables over time.

We can use this concept to choose $\mathbf{K}$ to satisfy the LQR (Linear Quadratic Regulator) problem - find $\mathbf{K}$ that minimises the cost function:

$$J = \int_0^\infty (x^T Q x + u^T R u)\, dt$$

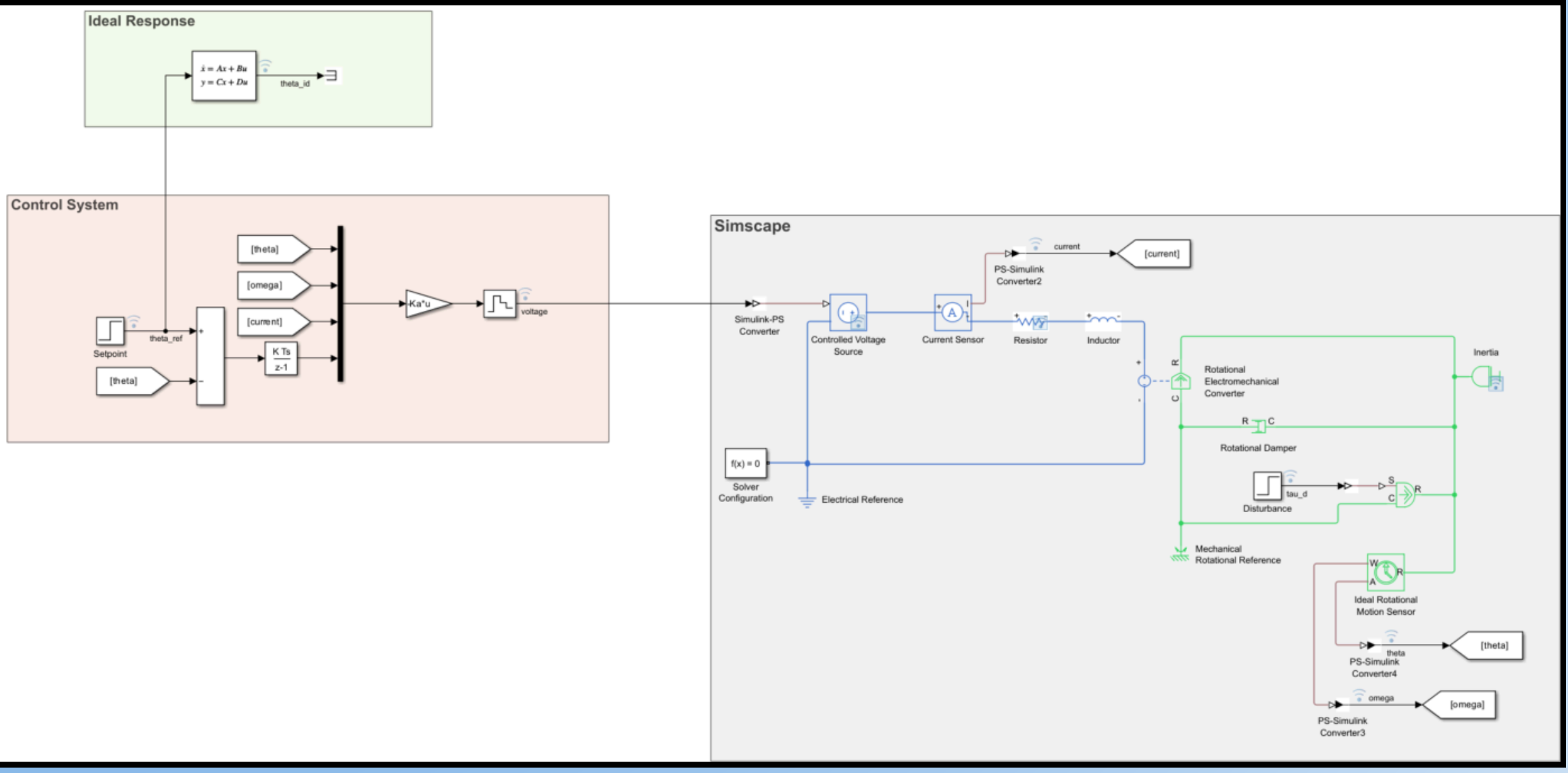
where:

- J is the scalar cost function to be minimized.
- x is the state vector of the system.
- Q is the state weighting matrix, which is symmetric and positive semi-definite.
- u is the control vector.
- R is the control weighting matrix, which is symmetric and positive definite.
- t represents time, and the integration is performed over the infinite horizon.

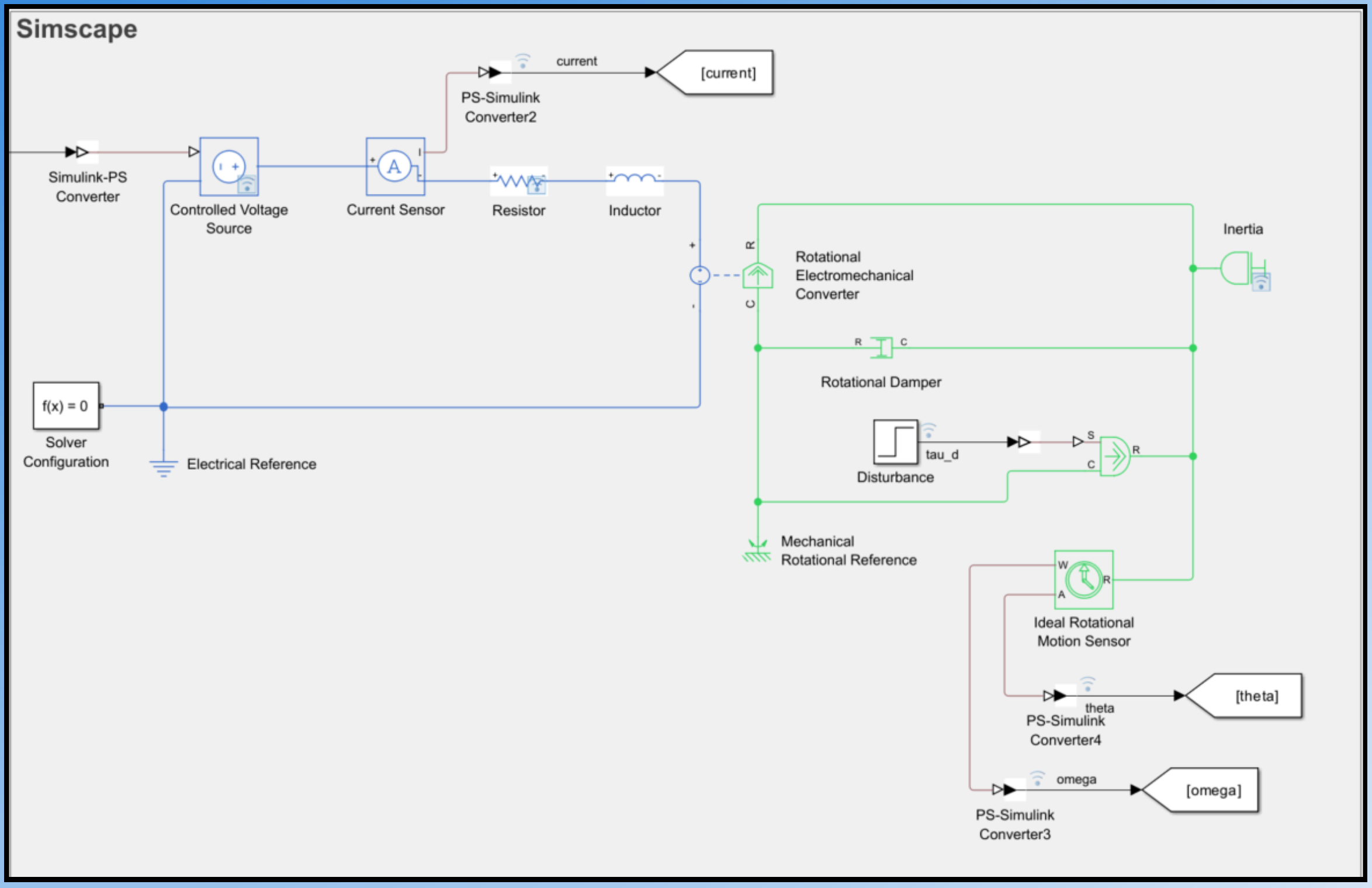
In our case, we can find the state feedback gain for the augmented system (K_a) using the Matlab command:

K_a = lqr(A_a, B_a, Q, R)

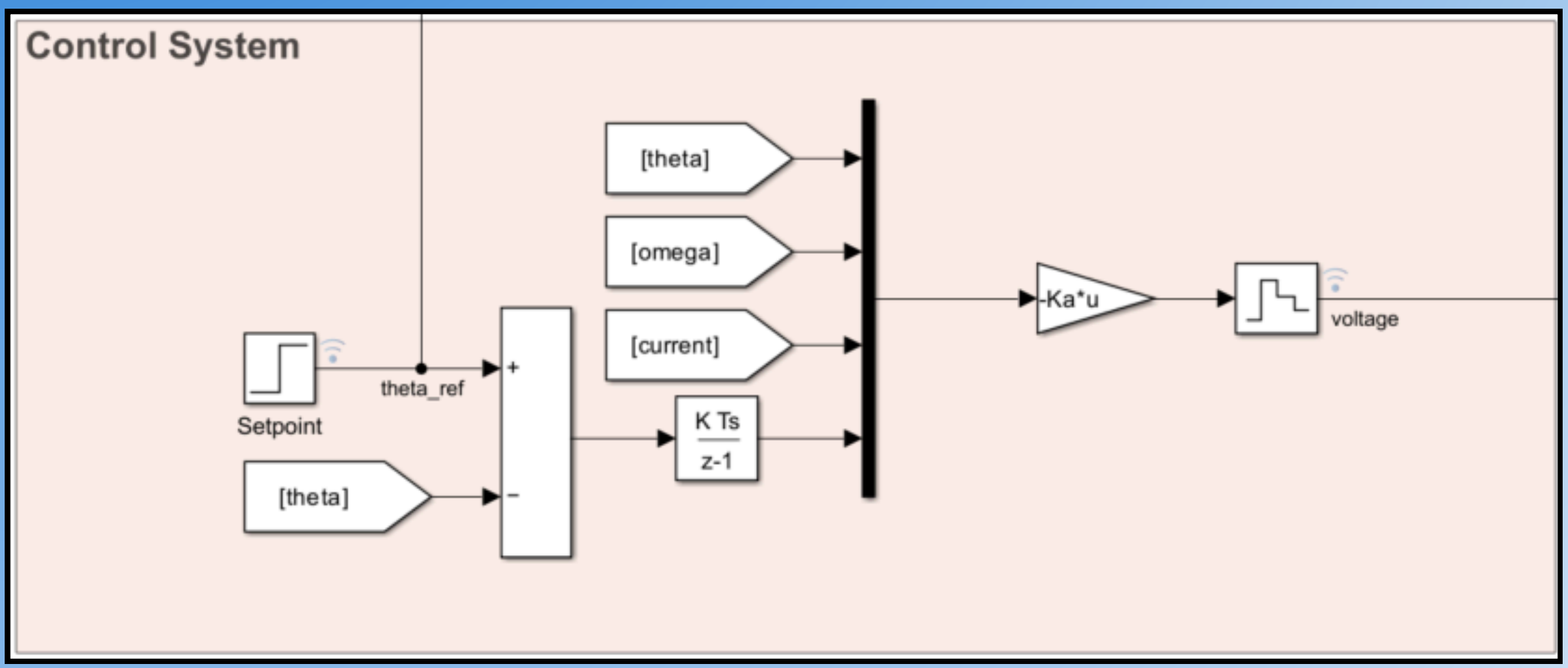
Simulink Model



DC Motor Model Simscape



Control System



Matlab – Init Code

```matlab
%% Init params
m = 0.1;
r = 0.05;
J = 0.5*m*r^2;
b = 0.0000095;
kt = 0.0187;
R = 0.6;
L = 0.35/1000;
ke = 0.0191;

%% Controller params

% Compute A, B, C, D matrices
A = [0 1 0; 0 -b/J kt/J; 0 -ke/L -R/L];
B = [0; 0; 1/L];
C = [1 0 0];
D = 0;

% Compute A, B, C, D matrices for augmented system
Aa = [A, zeros(3,1); -C, 0];
Ba = [B; -D];
Ca = [C 0];

% Compute state feedback gains using LQR for augmented system
Q = diag([1 1 1 100]);
R_ = 1;
Ka = lqr(Aa, Ba, Q, R_);

% Calculate ideal response subsystem
idResp = ss(Aa - Ba*Ka, [zeros(3,1); 1], Ca, D);

tSamp = 0.001;
```

Simulation

The following simulation shows the result where there is a step torque disturbance (0.1 Nm) injected at time 2.5 seconds.

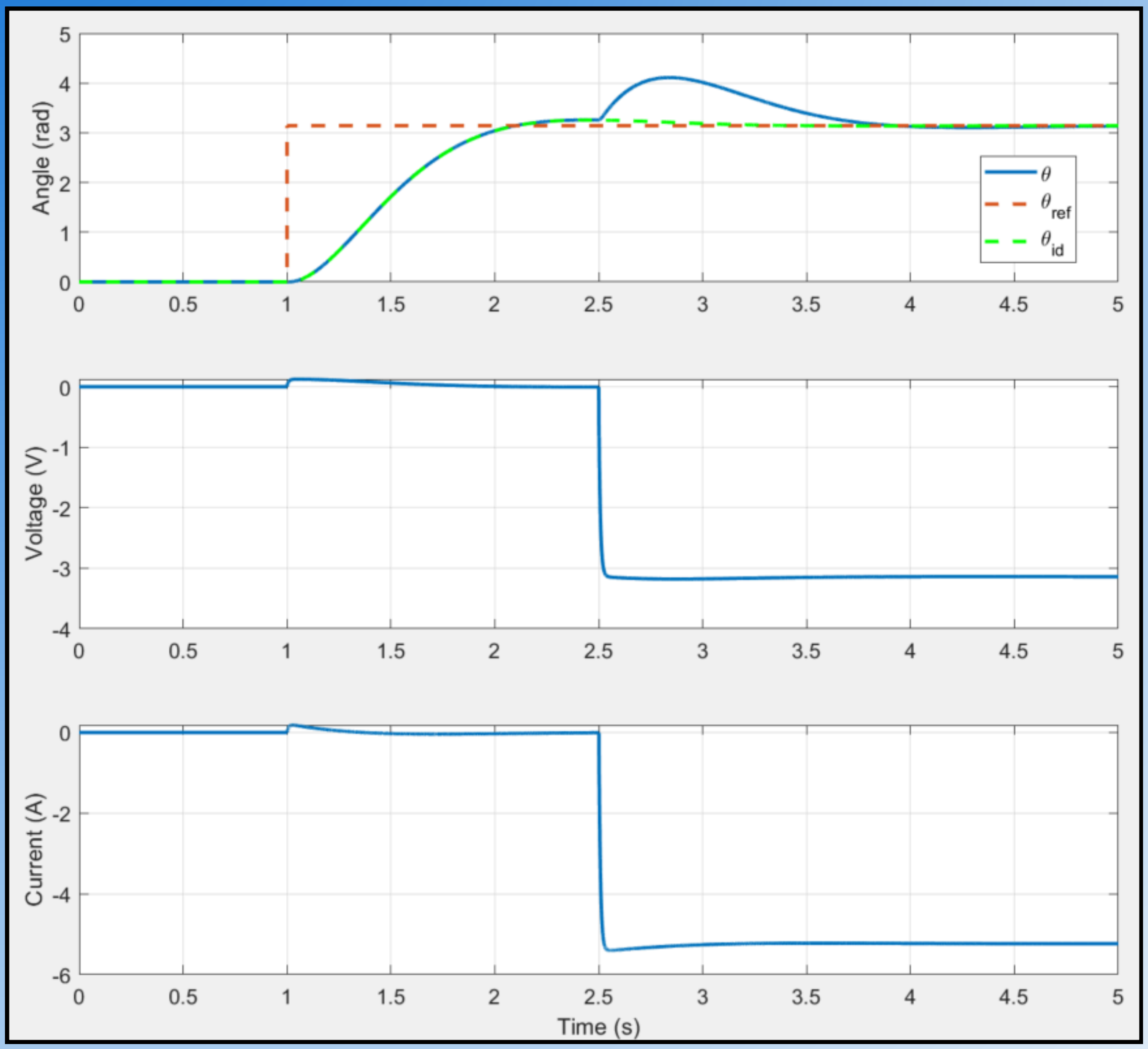
The disturbance is additive on the torque, hence the mechanical equation is:

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

The setpoint r changes from 0 rad to π rad at time 1 second.

The control system runs in discrete time, sample time = 0.001 seconds.

Result



Matlab – Plot Code

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

theta = out.logsout.get('theta').Values.Data;
t_theta = out.logsout.get('theta').Values.Time;

theta_ref = out.logsout.get('theta_ref').Values.Data;
t_theta_ref = out.logsout.get('theta_ref').Values.Time;

theta_id = out.logsout.get('theta_id').Values.Data;
t_theta_id = out.logsout.get('theta_id').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;

omega = out.logsout.get('omega').Values.Data;
t_omega = out.logsout.get('omega').Values.Time;


% Create the first figure
figure;

% Subplot for theta
subplot(3, 1, 1);
plot(t_theta, theta, 'LineWidth', 2);
hold on;
plot(t_theta_ref, theta_ref, '--', 'LineWidth', 2);
plot(t_theta_id, theta_id, 'g--', 'LineWidth', 2);
hold off;
%xlabel('Time (s)');
ylabel('Angle (rad)');
legend({'\theta', '\theta_{ref}', '\theta_{id}'}, 'FontSize', 12);
set(gca, 'FontSize', 12);
grid on;

% Subplot for voltage
subplot(3, 1, 2);
plot(t_voltage, voltage, 'LineWidth', 2);
%xlabel('Time (s)');
ylabel('Voltage (V)');
set(gca, 'FontSize', 12);
grid on;

% Subplot for current
subplot(3, 1, 3);
plot(t_current, current, 'LineWidth', 2);
xlabel('Time (s)');
ylabel('Current (A)');
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