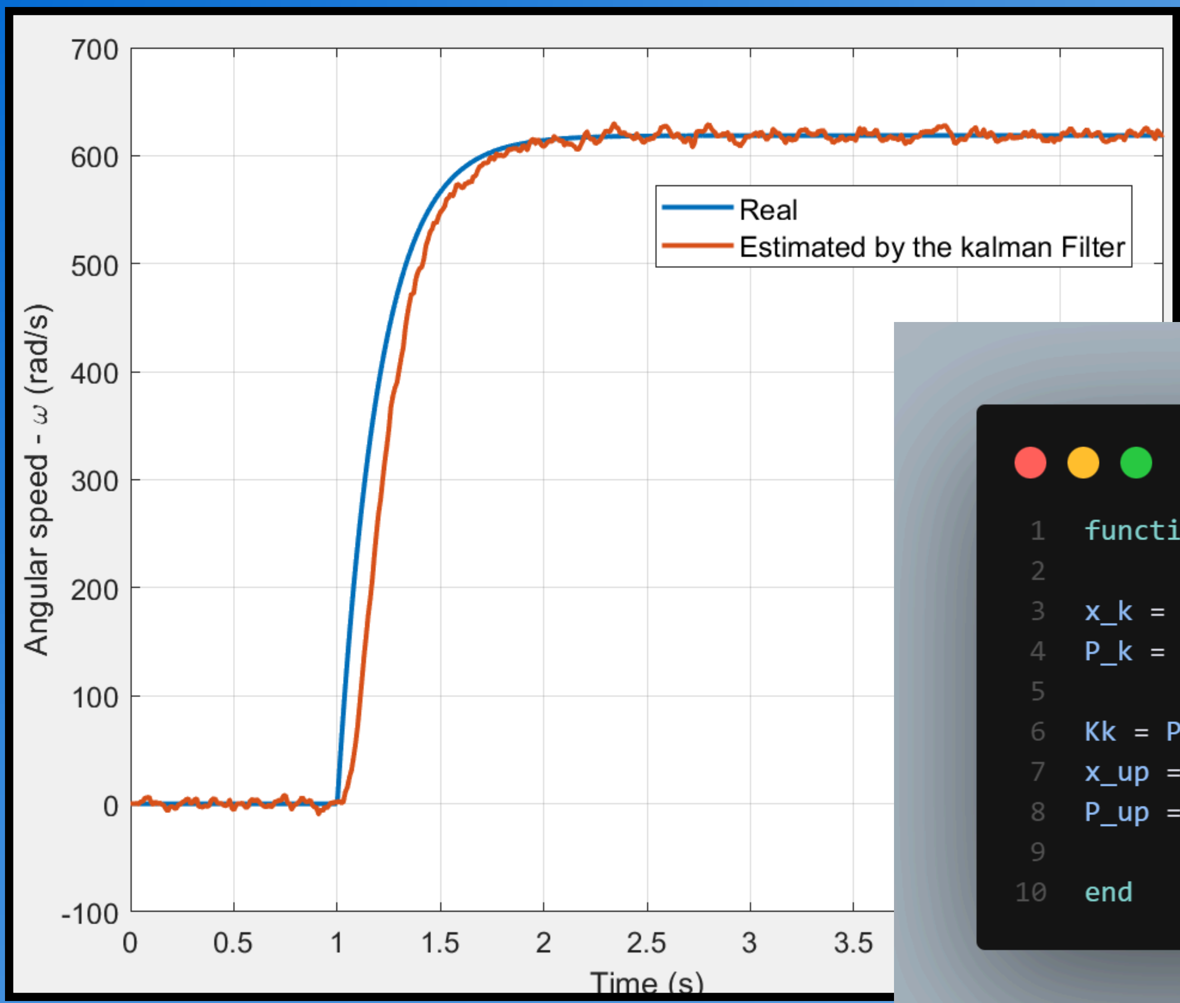
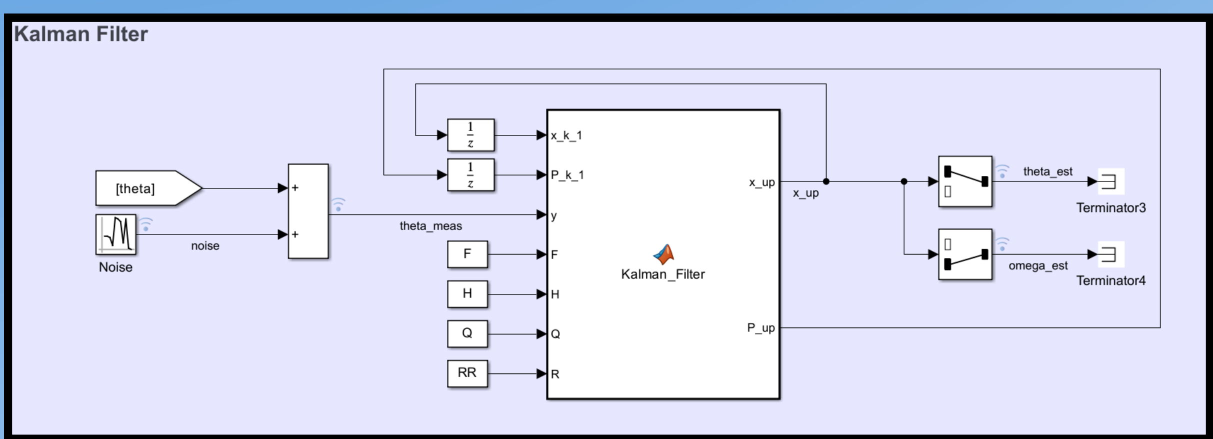
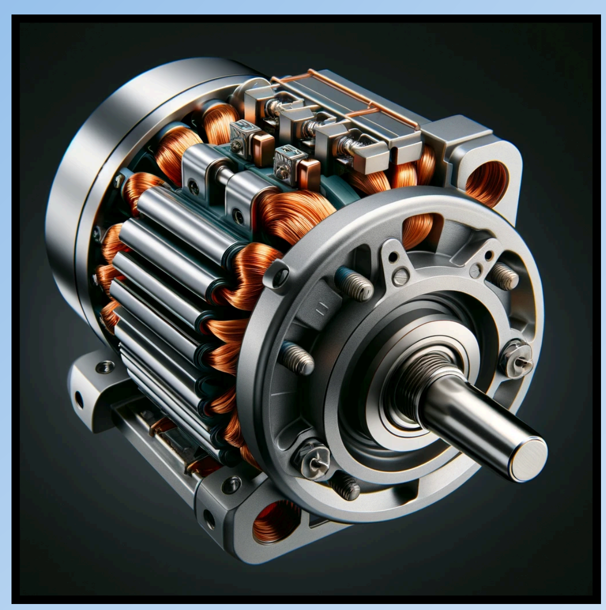
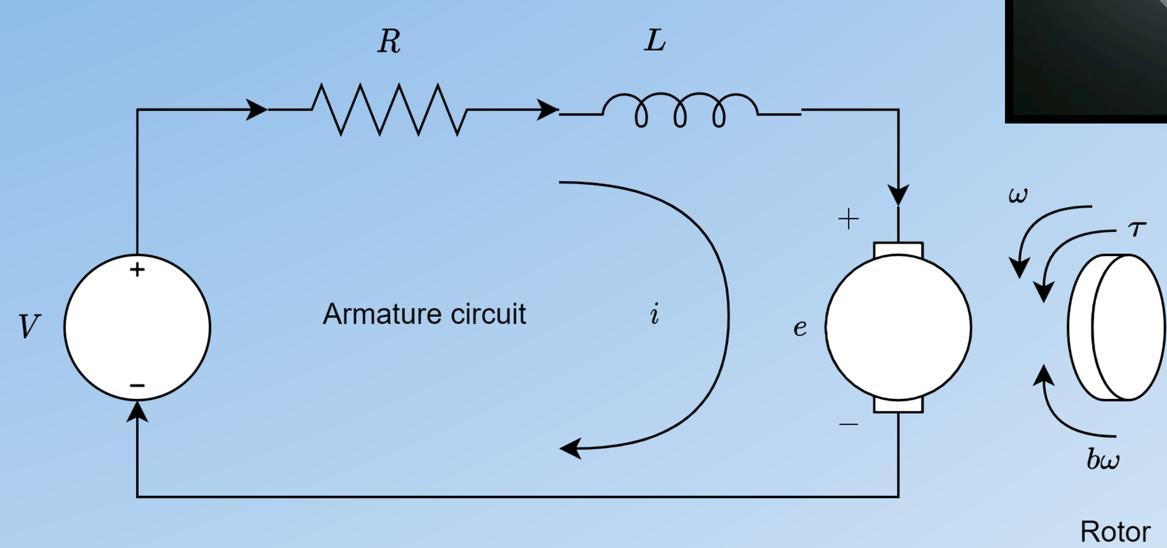



```
function [x_up, P_up] = Kalman_Filter(x_k_1, P_k_1, y, F, H, Q, R)

x_k = F*x_k_1;
P_k = F*P_k_1*F'+Q;            % Project the State Covariance Ahead

Kk = P_k*H'*inv(H*P_k*H'+R);   % Kalman Gain
x_up = x_k+Kk*(y-H*x_k);       % Update Measurement with measurement
P_up = P_k-Kk*H*P_k;           % Update the Error Covariance

end
```

How to Use a Kalman Filter to Estimate the Angular Velocity of a DC Motor







Code

https://github.com/simorxb/Kalman-Filter-DC-Motor

SIMONE BERTONI CONTROL LAB

Kalman Filter Model

The Kalman filter is one of the most important and common estimation approach that estimates an unknown state of a dynamic system from a series of noisy measurements.

The Kalman filter model assumes a linear dynamic system discretized:

$$x_k = Fx_{k-1} + Bu_k + w_k$$
$$z_k = Hx_k + v_k$$

where

$$w_k \sim \mathcal{N}(0, Q)$$
$$v_k \sim \mathcal{N}(0, R)$$

are the process and measurement noise, both drawn from a zero mean Gaussian noise, with covariance respectively Q and R.

Kalman Filter Implementation

The Kalman filter implementation calculates the predicted state and covariance estimate:

$$\hat{x}_{k|k-1} = Fx_{k-1|k-1} + Bu_k$$

$$\hat{P}_{k|k-1} = FP_{k-1|k-1}F^T + Q$$

the innovation and its covariance:

$$\tilde{y}_k = z_k - H\hat{x}_{k|k-1}$$

$$S_k = H\hat{P}_{k|k-1}H^T + R$$

the optimal Kalman gain:

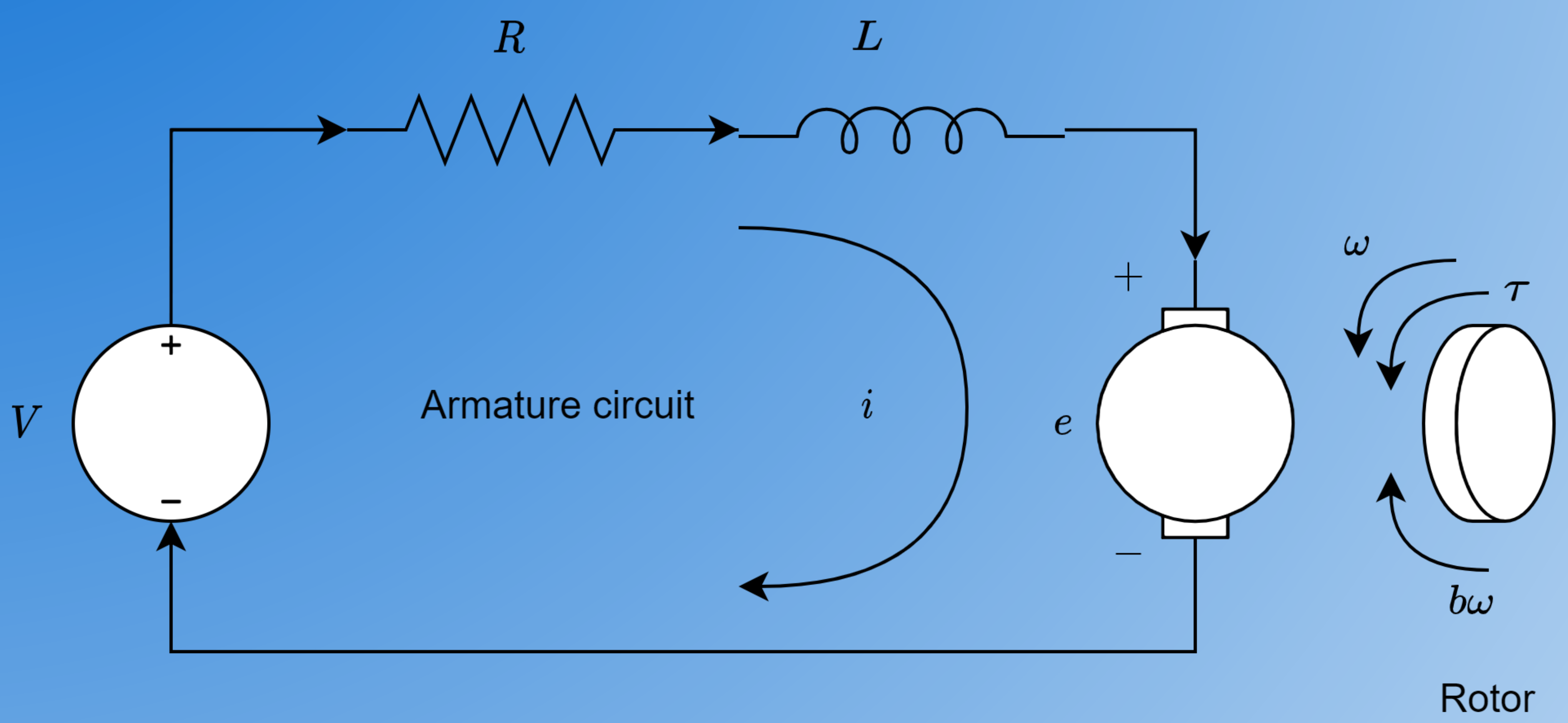
$$K_k = \hat{P}_{k|k-1}H^T S_k^{-1}$$

the updated state and covariance estimate:

$$x_{k|k} = \hat{x}_{k|k-1} + K_k\tilde{y}_k$$

$$P_{k|k} = (I - K_k H)\hat{P}_{k|k-1}$$

DC Motor



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:

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

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

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) = 0.35 mH

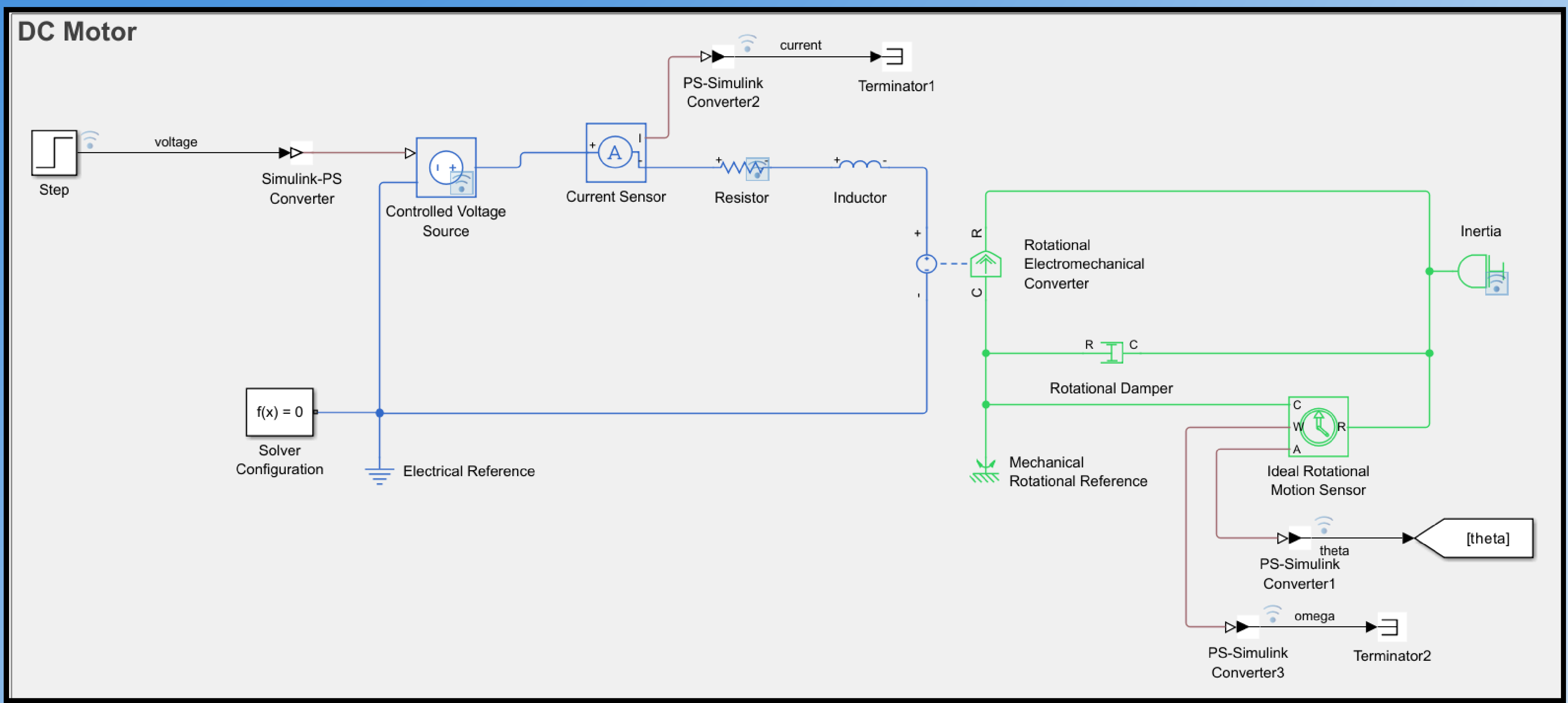
Damping factor (b) = 0.001 Nm/KRPM = 0.0000095 Nm/(rad/s)

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

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

```
%% Parameters

m = 0.1;            % Mass of the disc (kg)
r = 0.05;           % Radius of the disc (m)
J = 0.5*m*r^2;      % Moment of inertia of the disc (kg*m^2)
b = 0.0000095;      % Viscous friction coefficient (N*m*s)
kt = 0.0187;        % Torque constant (N*m/A)
R = 0.6;            % Armature resistance (Ohm)
L = 0.35/1000;      % Armature inductance (H)
ke = 0.0191;        % Back EMF constant (V*s/rad)
```

Simscape DC Motor Model



DC Motor
Model for Kalman Filter – 1

Here we want to design a Kalman Filter without assuming any knowledge of the DC Motor physics.

Hence we assume a kinematic model.

We measure the angular position every Δt seconds, and we want to estimate its angular velocity.

The position and velocity of the motor are described by the linear state space: $x_k = \begin{bmatrix} \theta \\ \omega \end{bmatrix}$, where θ: angular position and ω: angular velocity.

We assume that \alpha(k) (acceleration) is unknown and normally distributed with mean 0 and standard deviation σ_α. Then:

$$x_k = Fx_{k-1} + G\alpha_k$$

where $F = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix}$ and $G = \begin{bmatrix} \frac{1}{2}\Delta t^2 \\ \Delta t \end{bmatrix}$

DC Motor
Model for Kalman Filter - 2

Then going back to the nominal dynamic system:

$$x_k = Fx_{k-1} + Bu_k + w_k$$

where $Bu_k = 0$ under the assumption of no known control inputs.

Therefore

$$w_k \sim \mathcal{N}(0, Q)$$

where

$$Q = GG^T\sigma_\alpha^2 = \begin{bmatrix} \frac{1}{4}\Delta t^4 & \frac{1}{2}\Delta t^3 \\ \frac{1}{2}\Delta t^3 & \Delta t^2 \end{bmatrix}\sigma_\alpha^2$$

The noisy measurement of the angular position of the rotor is

$$\theta_k = Hx_k + v_k$$

where $H = \begin{bmatrix} 1 & 0 \end{bmatrix}$. v_k is normally distributed with mean 0 and standard deviation σ_θ. Hence $R = \sigma_\theta^2$.

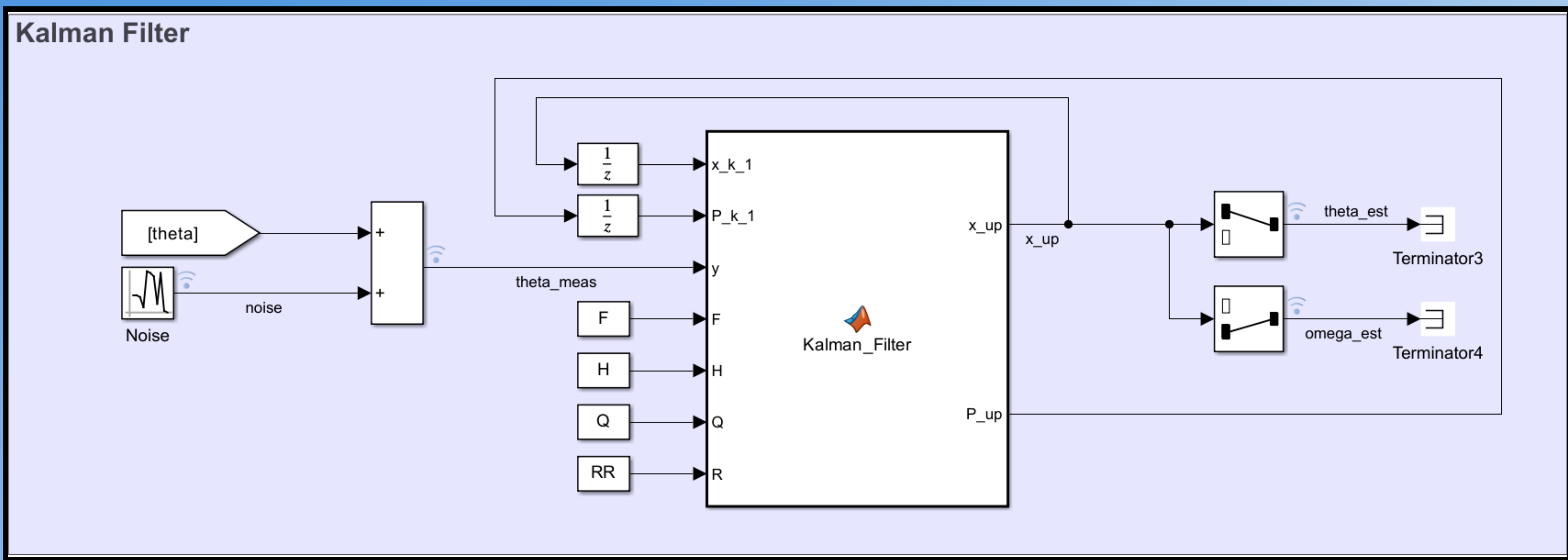
Matlab – Kalman Filter Initialisation

```matlab
%% Kalman Filter

T = 0.01;               % Sampling time (s)
H = [1 0];              % Measurement matrix
F = [1 T; 0 1];         % State transition matrix
G = [0.5*T^2; T];       % Input matrix

s_alpha = 300;          % Angular acceleration noise standard deviation
s_theta = 1;            % Measurement noise standard deviation

Q = G*G'*s_alpha^2; % Process noise covariance matrix
RR = s_theta^2;         % Measurement noise covariance

P0 = 0*eye(2,2);        % Initial error covariance matrix
x0 = [0; 0];            % Initial state estimate [theta; omega]
```

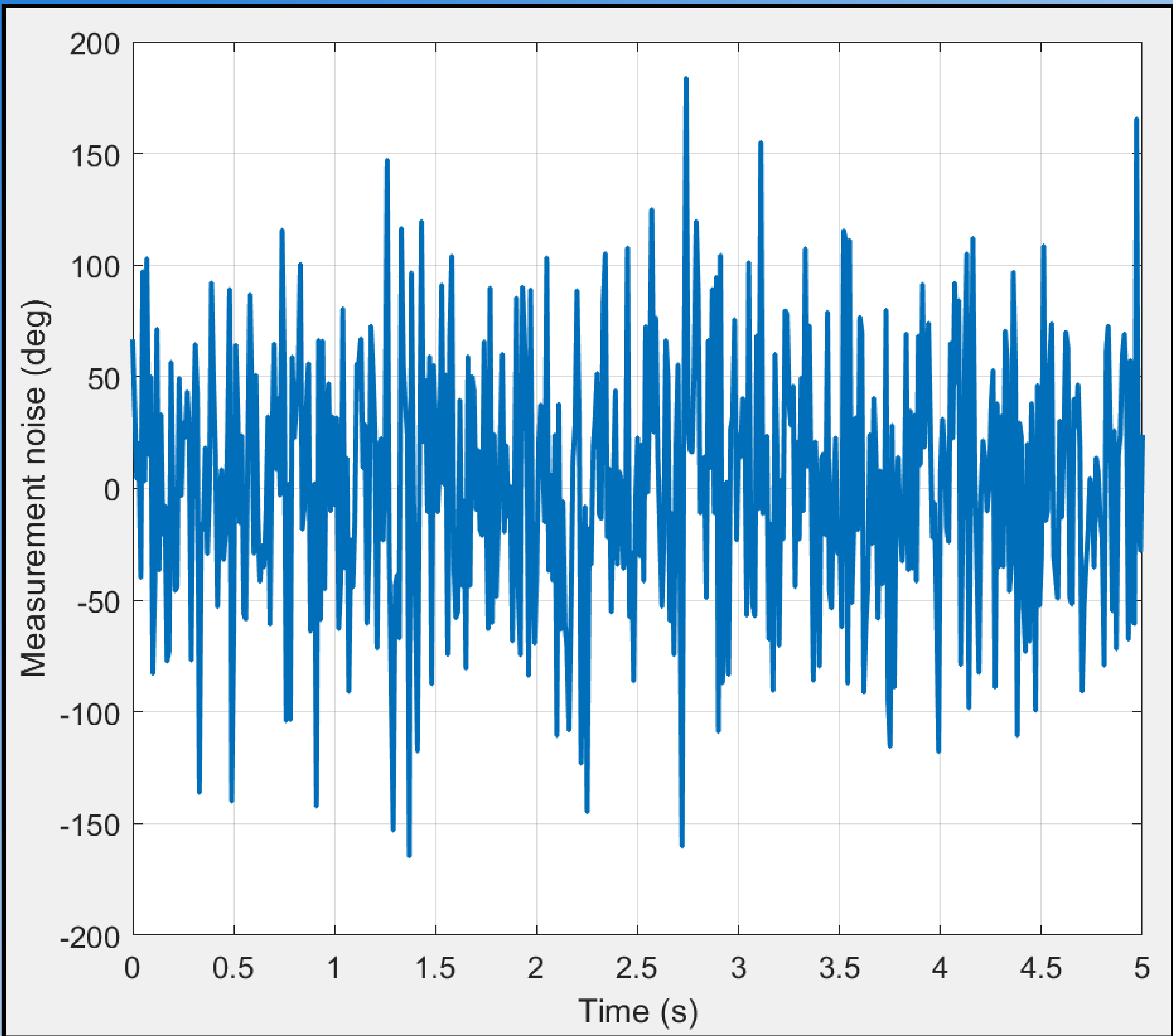
Matlab – Kalman Filter Implementation

```matlab
function [x_up, P_up] = Kalman_Filter(x_k_1, P_k_1, y, F, H, Q, R)

x_k = F*x_k_1;
P_k = F*P_k_1*F'+Q;                 % Project the State Covariance Ahead

Kk = P_k*H'*inv(H*P_k*H'+R);        % Kalman Gain
x_up = x_k+Kk*(y-H*x_k);            % Update Measurement with measurement
P_up = P_k-Kk*H*P_k;                % Update the Error Covariance

end
```

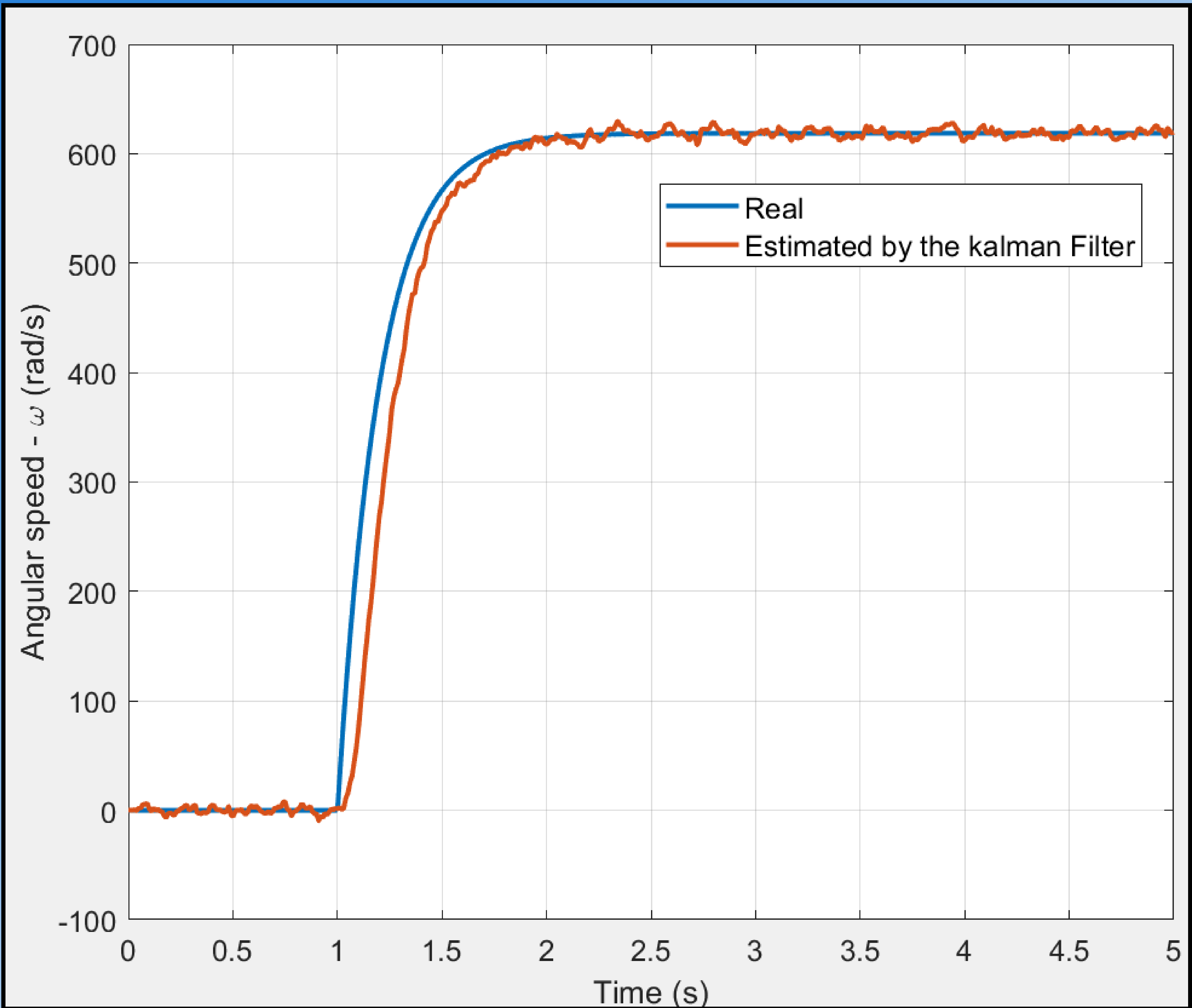
Simulink – Kalman Filter Implementation



Measurement Noise



Angular Velocity Estimation



PID Controller Course

https://simonebertonilab.com

Overall, a great course. worth recommending.

★★★★★ *May 7, 2024*

The course is exciting and engaging, covers most of the concepts and all the explanations are clear and easy to understand. it would be great if there was more explanation on the calculations and recommendations for additional resources. Overall excellent course, and I got to learn a lot.

Vikram Kolupula ✓ 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.

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 good sharing of experience

Romy Domingo Bompart Ballache

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