Linear State Estimators for a Load Estimation in a PMSM system

Barna Temesi

Abstract—This short design documentation presents the used system model, parameters and load observer. The used observers are a Luenberger observer and various different forms of the Kalman-filter. The estimated signal is then used for feed-forward load torque compensation to enhance the transient response of the system.

Index Terms: dynamic modelling of syncronous motors, *dq*-reference frame, PMSM, Load torque estimation, Luenberger observer, Kalman-filter, correlated noise, colored noise, Optimal state estimation

1 INTRODUCTION

As in every design process, after the problem statement, the system has to be modeled and analyzed. This starts with an overview of the motor parameters. Next, the motor voltage equations, which are already given in the dq-reference frame, are examined. The mechanical equation of the motor is also presented.

This documentation does not consider the FOC used in the system. The focus is strictly on the load torque estimators and feed-forward load torque compensation.

The basis of load torque estimation is my thesis [1] and a Luenberger load observer publication [5]. The Kalman-filter is mostly based on the marvelous book of: [6], but the original papers ([7], [8]) were also re-visited during this study. Thanks to Phil, the very neatly organised and dense tutorials found on the website of [9], helped with the first steps in the implementation of the algorithms.

2 MODEL OF THE SYSTEM

The motor in the scope of this design report, is an SPMSM. The load torque is generated using an IM machine which is connected to the SPMSM using a coupling.

Surface mounted PMSM means that the permanent magnets are located on the surface of the rotor. Due to this, the motor is non-salient, and also the reluctance path is equal on the d- and q-axis. This results in equal inductance on the d- and q-axis. For easier understanding, the machine inductance will be denoted as L_s [2].

$$L_d = L_q = L_s \tag{1}$$

The most important parameters of the motor and the other necessary system parameters are listed in table 1.
As it can be seen from the table, the motor has 4 pole pairs. Generally speaking, this means that the machine is more geared towards high-speed operation. In high-torque operation applications, like in the case of a steering motor, the number of poles might exceed a 100.
In simulations, the total system resistance will be used, which takes into account the resistance of every possible component in the setup [1].

Table 1: *System parameters, from previous projects such as*

Description	Notation	Value	Unit
Number of pole pairs	N_{pp}	4	-
Winding resistance	R_w	0.19	Ω
Total system resistance	R_s	0.268	Ω
q and d-axis inductance	L_m	2.2	mH
Rotor PM flux linkage	λ_{mpm}	0.12258	wB
Rated speed, SPMSM	$\omega_{m,rated}$	4500	rpm
Rated torque, SPMSM	$\tau_{m,rated}$	20	Nm
Rated power, SPMSM	$P_{m,rated}$	9.4	kW
Rated speed, IM	$\omega_{IM,rated}$	1400	rpm
Rated torque, IM	$\tau_{IM,rated}$	14	Nm
Rated power, IM	$P_{IM,rated}$	2.2	kW
Rated current, VSI	I_{VSI}	35	A
IM machine inertia	J_{IM}	0.0069	$kg \cdot m^2$
SPMSM machine inertia	J_{SPMSM}	0.0048	$kg \cdot m^2$
Total system inertia	J_{sys}	0.0146	$kg \cdot m^2$
Coulomb friction	C	0.2295	Nm
Viscous friction	B	0.0016655	N
Sampling frequency	f_s	5000	Hz

The motor voltage equations are shown in equation (2). Due to the assumption that the system is symmetrical and balanced, the zero term (v_0) is zero.

$$\begin{aligned} v_d &= R_s i_d + p\lambda_d - \omega_r \lambda_q \\ v_q &= R_s i_q + p\lambda_q + \omega_r \lambda_d \\ v_0 &= 0 \end{aligned} \tag{2}$$

In the abc-reference frame, the machine flux-linkages are dependent on position. Position-varying inductanes are now constant because the model is already transformed into the $dq0$-reference frame.

The stator $dq0$-reference frame is aligned with the rotor reference frame, which is naturally in the $dq0$-reference frame. The rotor d-axis is chosen to be aligned with the maximum flux density line at no load condition. The q-axis is always leading the d-axis by 90 degrees electric. This way, it is aligned with the minimum flux density line [1].

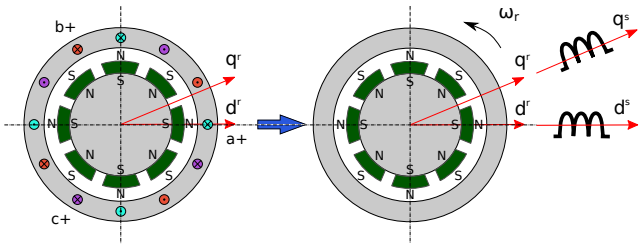

Figure 1: *Reference frame transformation from* ***abc*** *to dq. The structure of the motor is also shown. Inspiration: [4]*

The two d-axis is in line now. This is convenient because, it results in the d-axis and the q-axis flux-linkage as shown in equation (3).

$$\begin{aligned} \lambda_d &= (L_{ls} + L_{md})\, i_d + \lambda_{mpm} = L_d i_d + \lambda_{mpm} \\ \lambda_q &= (L_{ls} + L_{mq}) = L_q i_q \\ \lambda_0 &= 0 \end{aligned} \tag{3}$$

After substitution, the voltage equations may be rewritten as seen in equation (4).

$$\begin{aligned} v_d &= R_s i_d + p(L_d i_d + \lambda_{mpm}) - \omega_r L_q i_q \\ v_q &= R_s i_q + p(L_q i_q) + \omega_r (L_d i_d + \lambda_{mpm}) \end{aligned} \tag{4}$$

Where p is the differential operator $\frac{d}{dt}$. Differentiating the equation, keeping in mind that the derivative of a constant is zero, will result in the following.

$$\begin{aligned} v_d &= R_s i_d + L_d \cdot p i_d - \omega_r L_q i_q \\ v_q &= R_s i_q + L_q \cdot p i_q + \omega_r (L_d i_d + \lambda_{mpm}) \end{aligned} \tag{5}$$

In one more step, the homogeneous first-order differential equation of the system is acquired.

$$\begin{aligned} \frac{d}{dt} i_d &= -\frac{R_s}{L_d} i_d + \frac{1}{L_d} v_d + \omega_r \frac{L_q}{L_d} i_q \\ \frac{d}{dt} i_q &= -\frac{R_s}{L_q} i_q + \frac{1}{L_q} v_q - \omega_r \frac{L_d}{L_q} i_d - \frac{1}{L_q} \omega_r \lambda_{mpm} \end{aligned} \tag{6}$$

Equations (4) also contain the back-EMF voltage components which are important to highlight here:

$$\begin{aligned} e_d &= -\omega_r L_q i_q \\ e_q &= \omega_r (L_d i_d + \lambda_{mpm}) \end{aligned} \tag{7}$$

The governing torque equation can be derived from the equation of the input power of the windings. Simplifying this equation, using the attributions of the SPMSM machine, yields the following expression:

$$T_e = \frac{3}{2} N_{pp} \left(\lambda_d i_q - \lambda_q i_d\right) \tag{8}$$

$$T_e = \frac{3}{2} \frac{N_{poles}}{2} (\lambda_{mpm} i_q + (L_d - L_q) i_d i_q) \tag{9}$$

$$T_e = \frac{3}{2} N_{pp} \left(\lambda_{mpm} i_q\right) \tag{10}$$

Using Newton's second law, the mechanical equation of the system can be derived as shown in equation (11) [1].

$$T_e = j_{red} \frac{d\omega_m}{dt} + B_{m,red} \omega_m + T_{dist} \tag{11}$$

Where J is the total system inertia and T_{dist}, the disturbance torque, consists of the load torque and Coulomb friction. The total system inertia includes the inertia of both the IM and PMSM machine and also the coupling and fastening components between them.

The first term is related to the torque needed to accelerate the system without friction, the last two terms are related to the torque which is needed to overcome the viscous friction and the disturbance torque, respectively. Disturbance torque may be load torque, non-modelled friction (etc.) in the given system.

3 Load Torque Identification (LTID)

The various steps of the design of the observers are presented in this section.

Start by deriving the mechanical equation of the system using Newton's 2_{nd} law. This is in a slightly different form than shown before. This is presented in equations (12), (13).

$$j_{red} \dot{\omega}_m = \sum T_{sum} \tag{12}$$

$$\dot{\omega}_m = \frac{1}{j_{red}} \sum T_{sum} \tag{13}$$

Detail this, such as:

$$j_{red} \dot{\omega}_m = T_m - T_{load} - B_{m,red} \omega_m - T_{dist,residual} \tag{14}$$

Where, the actuator dynamics are shown in (15). The actuator dynamics were reduced to first-order. The time constant is τ_m.

$$\dot{T}_m = \frac{1}{\tau_m} (T_{m,ref} - T_m) \tag{15}$$

The modified system structure is depicted in figure 2. The feed-forward compensation helps with the transient response and disturbance rejection of the control system.

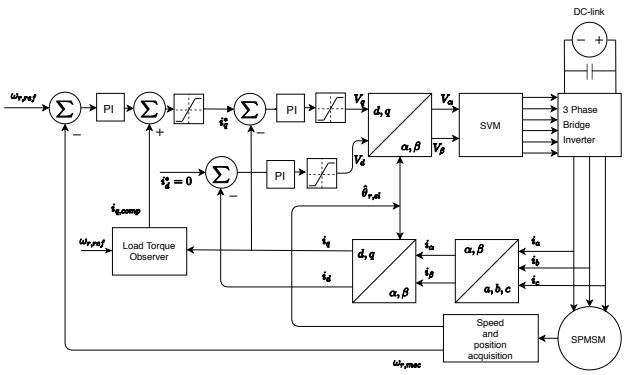

Figure 2: *Block diagram of the system with a load torque observer*

3.1 Luenberger Observer

A linear state observer is constructed based on the mechanical equation of the system. The plant in state-space form is presented in (16).

$$\begin{aligned} \dot{\boldsymbol{x}} &= A\boldsymbol{x} + B\boldsymbol{u} + Gw \\ \boldsymbol{y} &= C\boldsymbol{x} + D\boldsymbol{u} + \nu \end{aligned} \tag{16}$$

First, the state-space representation of the system is established in equations (17), (18). The load is already transformed to be a state. The input *u* is the actual motor torque T_m. The actuator dynamics are neglected for now.

$$\begin{bmatrix} \dot{\omega}_m \\ \dot{T}_{load} \end{bmatrix} = \begin{bmatrix} -\frac{B_{m,red}}{j_{red}} & -\frac{1}{j_{red}} \\ 0 & \tau_{load} \end{bmatrix} \begin{bmatrix} \omega_m \\ T_{load} \end{bmatrix} + \begin{bmatrix} \frac{1}{j_{red}} \\ 0 \end{bmatrix} T_m + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} w_{MotSpd} \\ w_{dist,load} \end{bmatrix} \tag{17}$$

$$\omega_m = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} \omega_m \\ T_{load} \end{bmatrix} + \nu_{MotSpd} \tag{18}$$

Where, τ_{load} is either -1/1000 or 0 (this is unconventional, but it is defined like this to ensure that τ_{load} can be set to 0 with ease), and $D = 0$.

Before designing the state observer, the observability of the second-order system has to be checked. The observability matrix is shown in equation (19).

$$\boldsymbol{Ob} = \begin{bmatrix} \boldsymbol{C} \\ \boldsymbol{CA} \end{bmatrix} \tag{19}$$

The matrix is full rank, therefore the system is observable. Utilizing linear control theory, a linear state observer can be designed [5], [6]. This is presented in equations (20).

$$\begin{aligned} \dot{\hat{\boldsymbol{x}}} &= A\hat{\boldsymbol{x}} + B\boldsymbol{u} + \boldsymbol{L}[\boldsymbol{y} - (C\hat{\boldsymbol{x}} + Du)] \\ \dot{\hat{\boldsymbol{x}}} &= (A - LC)\hat{\boldsymbol{x}} + [B \quad \boldsymbol{L}] \begin{bmatrix} u \\ y \end{bmatrix} \\ \hat{\boldsymbol{y}} &= C\hat{\boldsymbol{x}} \end{aligned} \tag{20}$$

The equation states that, by designing the observer gains ($\boldsymbol{L} = [l_1; l_2]$) to make the new system ($\boldsymbol{A} - \boldsymbol{LC}$) stable and sufficiently fast, the estimation error is driven to zero using the estimated speed as a feedback ($\tilde{\omega}_m = \omega_m - \hat{\omega}_m$). Sufficiently fast means that the observer has to be 3-5 times faster than the bandwidth of the control system [3].

$\boldsymbol{L}$ can be designed using the pole placement control approach. The closed-loop poles are placed at the desired place, and the new characteristic equation is calculated. By comparing the coefficients of the new and the old characteristic equations, the gains are found.

In this case, the poles of the new system are placed such as: $s_1 = [-50 - 50j]$, and $s_2 = [-50 + 50j]$. By doing so, the dampening of the system will be $1/\sqrt{2} \approx 0.707$, which will result in a stable (poles are located in the left-hand plane (LHP)), fast observer response with a small overshoot [3]. Pole placement is a valid method here because the observer is a mathematical construction, thus limiters (and physics) will not impose problems.

The overshoot of the observer may be reduced by increasing the damping of the observer.

For the given system, the calculated observer gains are the following:

$$\boldsymbol{L} = [100 \quad -73]' \tag{21}$$

The block diagram of the state observer is shown in figure 3.

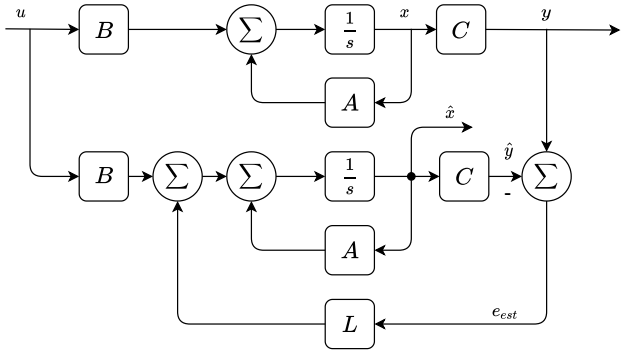

Figure 3: *Structure of the Luenberger Load Observer*

3.2 Discrete Luenberger Observer

The observer in the above discussed form is continuous which is obviously a problem. This subsection gives a simple solution to this problem. The estimator matrices are re-formulated using the now available observer gain $\boldsymbol{L}$.

Using figure 3, one can conclude that the new system matrices are the following:

$$\begin{aligned} A_{est} &= A - LC \\ B_{est} &= [B \quad \boldsymbol{L}] \\ C_{est} &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \\ D_{est} &= 0 \end{aligned} \tag{22}$$

Then, adopting the new matrices from (22), a new observer structure can be acquired. This is shown in figure 4. Discretization of the matrices using the MATLAB command 'c2dm' yields the observer in discrete form.

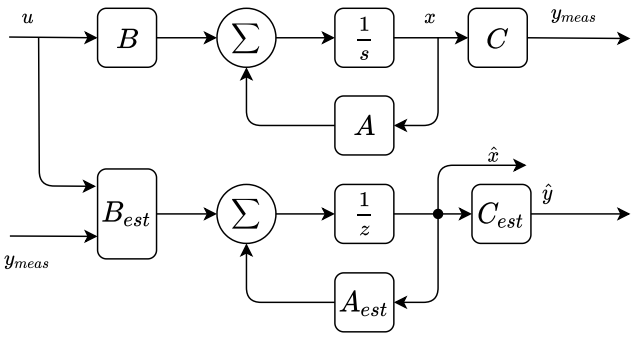

Figure 4: *Structure of the Discrete Luenberger Load Observer*

4 KALMAN-FILTER

In order to estimate the not measured states, and to filter the measured signals, a Kalman-filter is implemented. This is a popular option in industrial applications.
The Kalman-filter is the optimal (optimal in the sense of average estimation error) linear state estimator in case the noise is zero-mean, uncorrelated and Gaussian-distributed. Interestingly enough, the Kalman-filter is still the optimal linear filter in case the noise is not Gaussian-distributed [6].

In the given system, there is one measured state, the motor angle speed (ω_m). There are two objectives; the measured signal is corrupted with noise, therefore one objective is to filter this signal, effectively extracting the maximum available information out of it. The other objective is to estimate the load torque.

4.1 Discrete Kalman-filter

This section presents a simple discrete Kalman-filter in a similar form to as how it was introduced in the original paper written by Rudolf Kalman [7]. Other references used in this section are the following: [6] and [9].

First, the system in discrete state-space form is shown in equation (23) [6]. In practice, the system matrices shown in equations (17), (18) are discretized using MATLAB.

$$\begin{aligned} x_k &= F_{k-1}x_{k-1} + G_{k-1}u_{k-1} + w_{k-1} \\ y_k &= H_k x_k + v_k \end{aligned} \tag{23}$$

Where, the process noise and measurement noise are white, zero-mean, uncorrelated and have the following covariance matrices:

$$\begin{aligned} w_k &\sim (0, Q_k) \\ v_k &\sim (0, R_k) \\ E[w_k w_j^T] &= Q_k \delta_{k-j} \\ E[v_k v_j^T] &= R_k \delta_{k-j} \\ E[w_k v_j^T] &= 0 \end{aligned} \tag{24}$$

Where, δ_{k-j} is the Kronecker delta function, equals to 1 when $k = j$ and equals to 0 when $k \neq j$.

In the algorithm, the first step is to initialize the filter, to form $\hat{x}_0^+$ as the expected value of the initial state of x_0. The same is done for the initialization of the state estimation error covariance matrix (P_0^+). These are presented in equation (25).

$$\begin{aligned} \hat{x}_0^+ &= E(x_0) \\ P_0^+ &= E[(x_0 - \hat{x}_0^+)(x_0 - \hat{x}_0^+)^T] \end{aligned} \tag{25}$$

The following step is to propagate the state estimate. In literature, this is called the priori state estimate.

$$\hat{x}_k^- = F_{k-1}\hat{x}_{k-1}^+ + G_{k-1}u_{k-1} \tag{26}$$

Next is to update the state estimation error covariance matrix (P).

$$P_k^- = F_{k-1}P_{k-1}^+F_{k-1}^T + Q_{k-1} \tag{27}$$

Finally, the Kalman-gain can be computed as it is shown in equation (28) [6].

$$K_k = \frac{P_k^- H_k^T}{H_k P_k^- H_k^T + R_k} \tag{28}$$

Where, the denominator is the uncertainty in the measurement, and the numerator is the prior uncertainty.

The second step is to formulate the posteriori state estimate. This is also called the correction step.

$$\hat{x}_k^+ = \hat{x}_k^- + K_k(y_k - H_k\hat{x}_k^-) \tag{29}$$

At the end, the covariance matrix (P) is updated.

$$P_k^+ = (I - K_k H_k)P_k^- \tag{30}$$

It is expected that the covariance matrix (P) reduces overtime because the priori covariance matrix (P) is multiplied by a number less than one $(I - K_k H_k < 1)$. It is not a mathematician proof in any way, but one can have a feeling that due to the previous sentence, the objective of the minimization problem is fulfilled.

Let's treat the noise covariance matrices as design parameter. It can be shown that by increasing Q_k, the Kalman-gain is also increased. This is due to the increase in the state estimation error covariance matrix (P) as it was derived in equation (27). Also, decreasing R_k increases the Kalman-gain as well. This is clear from equation (28).
Increasing the Kalman-gain will put more emphasis on the measured data. In the correction step, the term $(y_k - H_k\hat{x}_k^-)$ is called the residual term, and a larger Kalman-gain will mean a larger second term in equation (29), thus a larger correction. On the other hand, by making the Kalman-gain zero or really close to zero, we essentially ignore any further measurements and say that the estimate is perfect.
These findings stand true for other variations of the Kalman-filter as well.

It was discussed in [6], chapter 10, that the Kalman-filter can be made more robust against modelling uncertainties, and uncertainties in Q_k, R_k by artificially increasing Q_k. At this point this was done by trial-and-error. It was shown in [6],

chapter 11, that the H_∞ estimator algorithm computes the optimal way of making the Kalman-filter more robust. As a note, the H_∞ estimator is not discussed in details in this report.

4.2 Kalman-Bucy filtering

In this section, the continuous-time Kalman-filter, also known as Kalman-Bucy-filter is discussed. The used sources for this section are the original publication [8], from 1961, the book [6] (chapter 8), and [9].

Suppose that the continuous-time system is given by the equations:

$$\begin{aligned} \dot{\boldsymbol{x}} &= A\boldsymbol{x} + B\boldsymbol{u} + w \\ \boldsymbol{y} &= C\boldsymbol{x} + \nu \\ w &\sim (0, Q_c) \\ v &\sim (0, R_c) \end{aligned} \tag{31}$$

Where, $w(t)$ and $v(t)$ are continuous-time white noise processes.
In the implementation, the system matrices shown in equations (17) and (18) are used again.

First, the filter is initialized.

$$\begin{aligned} \hat{x}(0) &= E[x(0)] \\ P(0) &= E[(x(0) - \hat{x}(0))(x(0) - \hat{x}(0))^T] \end{aligned} \tag{32}$$

The continuous-time Kalman-filter equations are given as [6]:

$$\begin{aligned} K &= PC^T R_c^{-1} \\ \dot{\hat{x}} &= A\hat{x} + Bu + K(y - C\hat{x}) \\ \dot{P} &= AP + PA^T - KRK^T + Q \end{aligned} \tag{33}$$

4.3 Steady-state Kalman filtering

It is often preferable to hard code a Kalman-gain into our implementation due to memory and computation effort restrictions in embedded systems. Therefore, K_k, P_k will not be updated at every time-step. This can be done under that assumptions that the underlying system is time-invariant, and the process and measurement noise covariances are also time-invariant [6], chapter 7.

The steady-state Kalman-filter is, strictly speaking, not optimal, but it approaches optimality in the limit as $k \to \infty$ [6]. It can be observed using the previous implementations of the filter that the Kalman gain converges to a given value over-time. Figure 5 clearly shows that the gains converged to a value after 50 and 500 time samples (recall that, the sampling time is: 0.2ms).

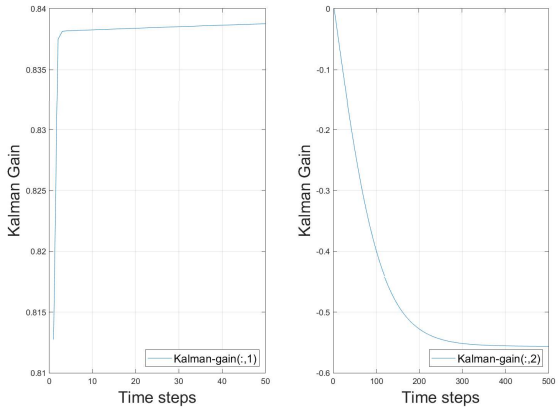

Figure 5: *Kalman gains captured during simulation*

Let's start by repeating the plant state-space representation here!

The plant in state-space form is presented in (34) and then detailed in equations (35), (36).

$$\begin{aligned} \dot{\boldsymbol{x}} &= A\boldsymbol{x} + B\boldsymbol{u} + Gw \\ \boldsymbol{y} &= C\boldsymbol{x} + D\boldsymbol{u} + \nu \end{aligned} \tag{34}$$

$$\begin{bmatrix} \dot{\omega}_m \\ \dot{T}_{load} \end{bmatrix} = \begin{bmatrix} -\frac{B_{m,red}}{j_{red}} & -\frac{1}{j_{red}} \\ 0 & \tau_{load} \end{bmatrix} \begin{bmatrix} \omega_m \\ T_{load} \end{bmatrix} + \begin{bmatrix} \frac{1}{j_{red}} \\ 0 \end{bmatrix} T_m + \\ + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} w_{MotSpd} \\ w_{dist,load} \end{bmatrix} \tag{35}$$

$$\omega_m = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} \omega_m \\ T_{load} \end{bmatrix} + \nu_{MotSpd} \tag{36}$$

Where, τ_{load} is either -1/1000 or 0, and $D = 0$.
Also, the unbiased process noise w and measurement noise ν have the following covariances:

$$\begin{aligned} w &\sim (0, Q_c) \\ v &\sim (0, R_c) \end{aligned} \tag{37}$$

We have to have priori knowledge of these values.

The objective is to minimize the steady-state estimation error covariance as shown in equation (38).

$$P = \min_{t \to \infty} E\left([\boldsymbol{x} - \hat{\boldsymbol{x}}][\boldsymbol{x} - \hat{\boldsymbol{x}}]^T\right) \tag{38}$$

The Kalman gain is acquired by solving an Algebraic Riccati equation. The used algorithm will yield a stable estimator, however, as a last step, it is worth checking the eigenvalues of $(\boldsymbol{A} - \boldsymbol{KC})$.
After computing the Kalman-gain, the previously introduced state-observer structure can be used again. The only change is that, now, the observer gain is replaced by the Kalman-filter gain.

The structure of the Kalman-filter can be seen in figure 6.

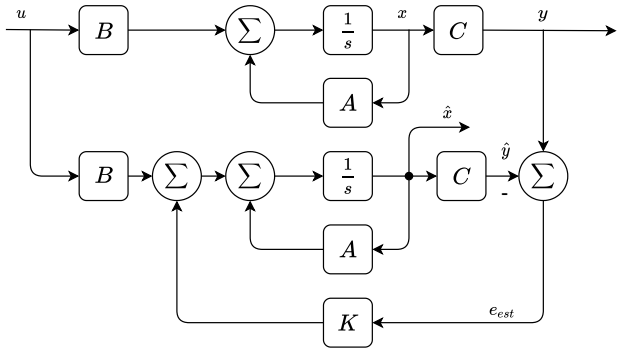

Figure 6: *Structure of the Kalman-filter*

4.4 Discrete Steady-state Kalman filtering

The previously introduced discretization of the Luenberger observer (see section 3.2) can be re-used here too. This is shown in figure 7. The system matrices are calculated such as:

$$\begin{aligned} A_{dKF} &= A - KC \\ B_{dKF} &= [B \quad K] \\ C_{dKF} &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \\ D_{dKF} &= 0 \end{aligned} \tag{39}$$

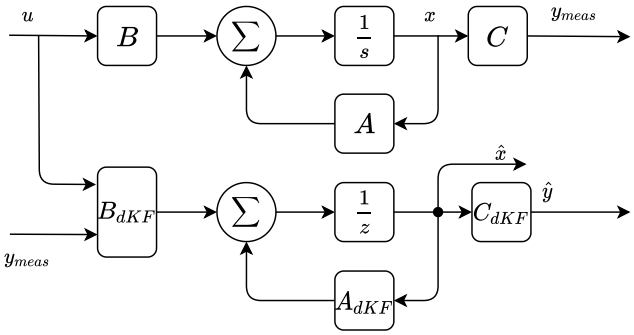

Figure 7: *Structure of the Discrete Kalman-filter*

4.5 Correlated process and measurement noise

In previous designs, it was assumed that the process and measurement noise are uncorrelated. This condition is often not satisfied. Thus, the defined cross-covariance matrix containing information of this is not zero. The topic of correlated noise was discussed in details in reference [6], section 2.7. The algorithm found there was implemented and shared on the Mathworks File Exchange website by michaelB brost [10].
The Kalman-filter algorithm that also takes into account the correlation in the noise was discussed in reference [6], section 7.1.

Let's re-visit the system presented in discrete form!

$$\begin{aligned} x_k &= F_{k-1}x_{k-1} + G_{k-1}u_{k-1} + w_{k-1} \\ y_k &= H_k x_k + v_k \end{aligned} \tag{40}$$

Where, the process noise and measurement noise are white, zero-mean, have the following covariance matrices:

$$\begin{aligned} w_k &\sim (0, Q_k) \\ v_k &\sim (0, R_k) \\ E[w_k w_j^T] &= Q_k \delta_{k-j} \\ E[v_k v_j^T] &= R_k \delta_{k-j} \\ E[w_k v_j^T] &= M_k \delta_{k-j+1} \end{aligned} \tag{41}$$

The cross covariance matrix is given by M_k. Previously, this was assumed to be 0. Now, the derivation of the filter equations has to be re-done including M_k. This was detailed in [6]. Here, a summary of the algorithm is presented.
The Kalman-filter is initialized such as:

$$\begin{aligned} \hat{x}_0^+ &= E(x_0) \\ P_0^+ &= E[(x_0 - \hat{x}_0^+)(x_0 - \hat{x}_0^+)^T] \end{aligned} \tag{42}$$

The next steps are given in the same fashion as it was presented in section 4.1.

$$P_k^- = F_{k-1}P_{k-1}^+F_{k-1}^T + Q_{k-1} \tag{43}$$

$$K_k = \frac{P_k^- H_k^T + M_k}{H_k P_k^- H_k^T + H_k M_k + M_k^T H_k^T + R_k} \tag{44}$$

$$\hat{x}_k^- = F_{k-1}\hat{x}_{k-1}^+ + G_{k-1}u_{k-1} \tag{45}$$

$$\hat{x}_k^+ = \hat{x}_k^- + K_k(y_k - H_k\hat{x}_k^-) \tag{46}$$

$$\begin{aligned} P_k^+ &= (I - K_k H_k)P_k^-(I - K_k H_k)^T + \\ &\quad + K_k(H_k M_k + M_k^T H_k^T + R_k)K_k^T - M_k K_k^T - K_k M_k^T \\ &= P_k^- - K_k(H_k P_k^- + M_k^T) \end{aligned} \tag{47}$$

Setting M_k to be 0, the algorithm from section 4.1 is acquired, therefore the above discussed algorithm is a generalized version.

When using the steady-state Kalman-filter method (section 4.3), MATLAB offers the option to define an *N* noise cross covariance matrix and use it for the Kalman-gain computation.

$$N = E\{w\nu^T\} \tag{48}$$

Which is the following for the investigated system:

$$N = \begin{bmatrix} w_{MotSpd} \cdot \nu_{MotSpd} \\ w_{dist,load} \cdot \nu_{MotSpd} \end{bmatrix} \tag{49}$$

4.6 Colored process and measurement noise

It was also assumed in previous designs that the process noise and measurement noise found in the system is white. This is often violated as well. The 'what is colored noise' question was answered in [6], section 2.6, and the modified Kalman-filter algorithm was detailed in [6], section 7.2. There are various different types of colored noise in the literature. Only pink noise will be considered in this section. Pink noise is low-pass filtered white noise.
The standard steady-state Kalman-filter algorithm may be applied (as it was shown in section 4.3), however, the order of the system has to be increased. This new system is called augmented system, because the number of states is increased.

Pink noise can be generated by low-pass filtering of white noise such as:

$$\dot{\nu} = \tau_{noise}\nu + \zeta \tag{50}$$

Where, ζ is white noise, $\tau_{noise} = -\frac{1}{0.0006}$ is the inverse of the time constant of the LPF (this is unconventional, but it is defined like this to ensure that τ_{noise} can be set to 0 with ease) and ν is the colored noise.

Let's re-visit the system in state-space form!

$$\begin{aligned} \dot{\boldsymbol{x}} &= A\boldsymbol{x} + B\boldsymbol{u} + Gw \\ \boldsymbol{y} &= C\boldsymbol{x} + D\boldsymbol{u} + \nu \end{aligned} \tag{51}$$

$$\begin{bmatrix} \dot{\omega}_m \\ \dot{T}_{load} \end{bmatrix} = \begin{bmatrix} -\frac{B_{m,red}}{j_{red}} & -\frac{1}{j_{red}} \\ 0 & \tau_{load} \end{bmatrix} \begin{bmatrix} \omega_m \\ T_{load} \end{bmatrix} + \begin{bmatrix} \frac{1}{j_{red}} \\ 0 \end{bmatrix} T_m + \\ + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} w_{MotSpd} \\ w_{dist,load} \end{bmatrix} \tag{52}$$

$$\omega_m = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} \omega_m \\ T_{load} \end{bmatrix} + \nu_{MotSpd} \tag{53}$$

This is a simple system, only the motor speed is measured using a sensor. The actuator dynamics are neglected for now, thus the current sensors are assumed to be perfect.

Assume that the motor speed measurement noise is colored and have the following equation:

$$\dot{\nu}_{MotSpd} = \tau_{noise}\nu_{MotSpd} + \zeta_{MotSpd} \tag{54}$$

The augmented system model can be formulated as shown in equations (55), (56).

$$\begin{bmatrix} \dot{\omega}_m \\ \dot{T}_{load} \\ \dot{\nu}_{MotSpd} \end{bmatrix} = \begin{bmatrix} -\frac{B_{m,red}}{j_{red}} & -\frac{1}{j_{red}} & 0 \\ 0 & \tau_{load} & 0 \\ 0 & 0 & \tau_{noise} \end{bmatrix} \begin{bmatrix} \omega_m \\ T_{load} \\ \nu_{MotSpd} \end{bmatrix} + \\ + \begin{bmatrix} \frac{1}{j_{red}} \\ 0 \\ 0 \end{bmatrix} T_m + \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} w_{MotSpd} \\ w_{dist,load} \\ \zeta_{MotSpd} \end{bmatrix} \tag{55}$$

$$y = \begin{bmatrix} 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} \omega_m \\ T_{load} \\ \nu_{MotSpd} \end{bmatrix} + \tilde{\nu}_{MotSpd} \tag{56}$$

Then, the Kalman-filter is designed using the augmented system equations. The procedure of the design is exactly the same as it was discussed in sections: 4.3 and 4.4. An important thing to note here is that the measurement noise of the motor speed is converted into process noise. Now, ζ_{MotSpd} is the design parameter related to this.

After discretizing the system found in equations (55), (56), a Kalman-filter, similar to the one presented in section 4.1, can be realized. This means that F_k, G_k, H_k will be modified and Q_k, R_k will be adjusted as well. Reference [6] also presents a method that does not involve the augmentation of the system, instead the measurement equation is modified.
A steady-state version of this algorithm can be designed as well.

5 Simulation Results

This section shows and discusses the simulation results of the modeled load estimators defined in Sections 3 and 4. The base of all simulations and thus experiments is the system introduced in Section 2.
This is by no means a comprehensive study of the estimators. Only the Luenberger-observer (found in section 3.2) and the basic Kalman-filter (presented in section 4.4) are compared with each other.

The gains of the FOC PI controllers are held constant unless noted otherwise. The parameters for those two controllers are shown in table 2 and they are chosen based on report [1].

Table 2: *System parameters, from previous projects such as*

Description	Notation	Value	Unit
q-axis proportional gain	Kp_q	3.8	-
q-axis integral gain	Ki_q	463	-
d-axis proportional gain	Kp_d	3.8	-
d-axis integral gain	Ki_d	463	-
Speed proportional gain	Kp_{speed}	0.4	-
Speed integral gain	Ki_{speed}	1.2	-
Anti-windup gain	$K_{as,PM}$	5	-

First of all, the estimated load signals are compared in figure 8. Then, the responses are compared in better detail in figure 9. It is worth looking at the eigenvalues of the observers $(A - L \cdot C)$. These are the next; Luenberger-observer poles: $-50 \pm 50i$, Kalman-filter poles: $-58.59 \pm 58.45i$. Of course, it was my design choice to place the poles of the Luenberger-observer as such and make the observer slower than the Kalman-filter.

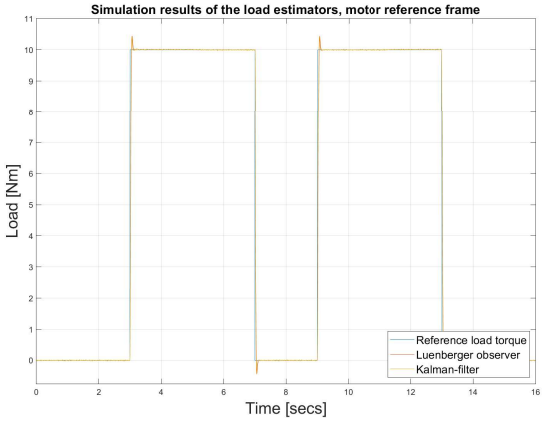

Figure 8: *Comparison of the estimated load signals*

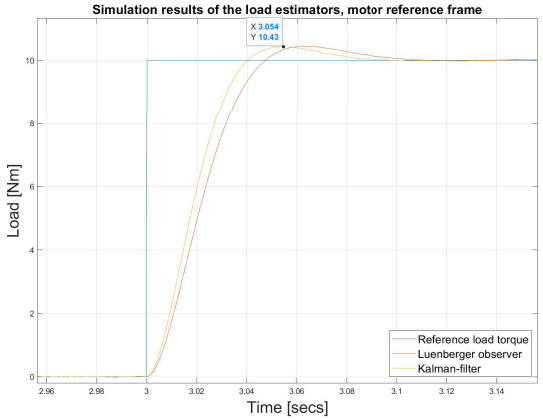

Figure 9: *Comparison of the estimated load signals - zoomed-in*

Next, the estimated load torque of the Kalman-filter is used as in feed-forward compensation. The speed controller was intentionally re-tuned to have pretty bad performance (low bandwidth, not sufficient disturbance rejection) and slow settling-time.

It can be clearly seen when comparing figures 11 and 13 that the transient response is greatly improved when the compensation was turned on. The speed drop and the settling time were reduced.
On the other hand, the response was more oscillatory and in some applications, this much of an overshoot is not preferred. Due to the fact that this is pretty general system (so, no strict requirements), the results were accepted.

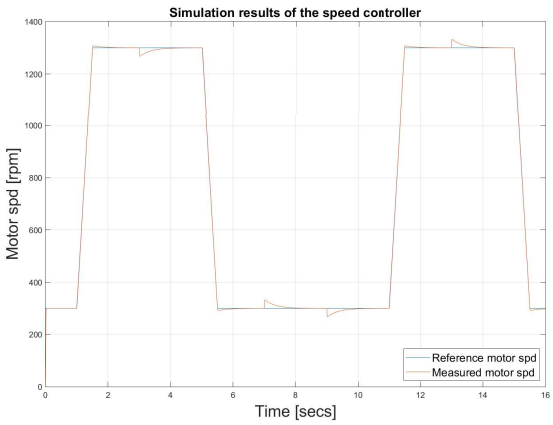

Figure 10: *Speed ctrl, transient response in simulation, feed-forward compensation disabled*

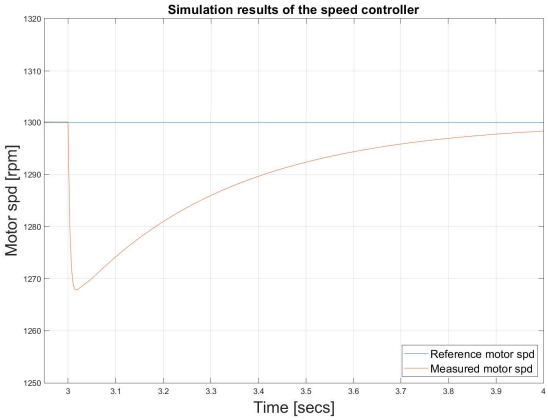

Figure 11: *Speed ctrl, transient response in simulation, zoomed-in, feed-forward compensation disabled*

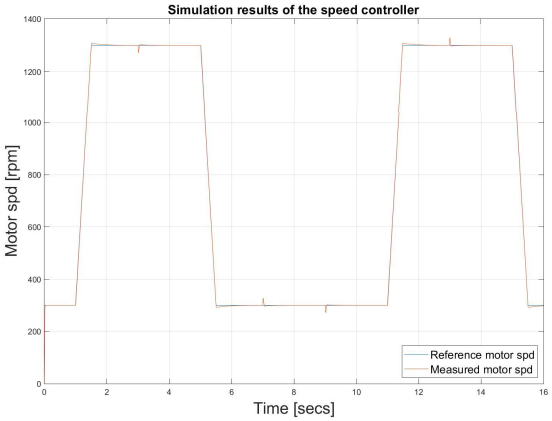

Figure 12: *Speed ctrl, transient response in simulation, feed-forward compensation enabled*

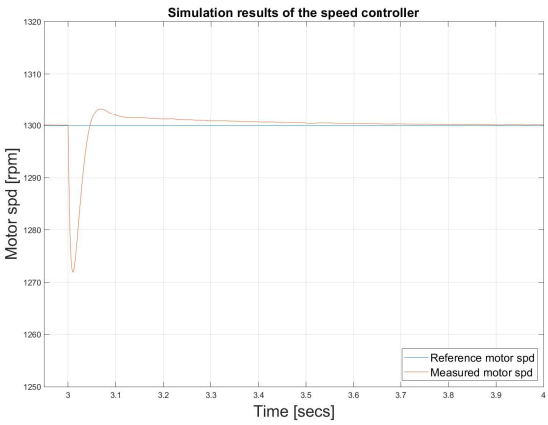

Figure 13: *Speed ctrl, transient response in simulation, zoomed-in, feed-forward compensation enabled*

6 CONCLUSION

The motivation for this study was to implement the Kalman-filter using time-varying Kalman gain. In practical applications, often the steady-state version is used and I was curious of the other forms of the Kalman-filter. This will be really useful during the study of nonlinear Kalman-filters because the steady-state version cannot be used then.

Both estimators performed well, although the tuning was not done thoroughly.
Of course, in an industrial application the Kalman-filter (or some form of a Kalman-filter) is preferred because its an optimal linear estimator, it minimizes the estimation error covariance matrix.

Nonetheless, the noise found in the system has to be analysed (correlated?, colored?) extensively in the early stages of the design process. After careful consideration, an estimator can be chosen that fits the problem very well.

As a last remark, often the linear model of a given system is not sufficiently good (for example: PMSM voltage equations in α, β-reference frame or an electro-mechanical system which takes nonlinear friction into account), therefore a nonlinear model has to be used. In these cases, nonlinear Kalman-filters may be used. Extended Kalman-filters (reference [6], section 13), unscented Kalman-filters (reference [6], section 14) and particle filters (reference [6], section 15) will be investigated in a later study.

REFERENCES

[1] B. Temesi, U. G. Gautadottir, *Sensorless Control of PMSM Drive Using Sliding-Mode-Observers* AAU, Denmark, 2020 Master's thesis.

[2] D. Wilson, *Motor Control Compendium*, 1st-ed . 2011

[3] C. L. Phillips, and R. D. Harbor, *Feedback Control Systems* 4th Edition. New Jersey: Prentice Hall, 2000, ISBN: 0-13-949090-6.

[4] K. Lu, *Control of Electrical Drive Systems and Converts Lecture 1 Slides*, 1st-ed . pp. 1-30, 2019

[5] Z. Kuang, B. Du, S. Cui, and C. C. Chan, *Speed Control of Load Torque Feedforward Compensation Based on Linear Active Disturbance Rejection for Five-Phase PMSM* in IEEE Access, vol. 7, pp. 159 787–159 796, 2019, ISSN:21693536. DOI: 10.1109/ACCESS.2019.2950368.

[6] Dan Simon, *Optimal State Estimation* 2006, John Wiley and Sons, Inc. ISBN: 13-978-0-471-70858-2

[7] R.E. Kalman, *A New Approach to Linear Filtering and Prediction Problems* ASME Journal of Basic Engineering, 82, pp. 35-45 (March 1960).

[8] R.E. Kalman, R.S. Bucy *New Results in Linear Filtering and Prediction Theory* ASME Journal of Basic Engineering, 83, pp. 95108 (March 1961).

[9] Phil Goddard, *Kalman Filters - Theory and Implementation* URL: https://www.goddardconsulting.ca/kalman-filter.html. Retrieved: March 13, 2022.

[10] michaelB brost, *Correlated Gaussian noise* URL: https://www.mathworks.com/matlabcentral/fileexchange/21156-correlated-gaussian-noise, MATLAB Central File Exchange. Retrieved: March 30, 2022.