Solution to Final Computer Exam

Abdur Rafay, 6920683

I. Introduction

Model Predictive Control (MPC) is a highly effective tool that sets up an optimization problem using a plant model to predict the behaviour of the plant and compare it with the reference trajectory in attempt to find the control input such that the cost function is minimized. MPC uses a step by step input utilization which is a good technique to counter model deviations and external disturbances. The advantage of MPC is that it can incorporate the input and state constraints in the model as well as can perform dynamic state and output reference tracking. The overall model can be condensed to an optimization problem which can be easily solve by Quadratic Programming solvers.

The control model of a induction motor is represented with continuous time LTI state space model. System matrix $\boldsymbol{A}$, input matrix $\boldsymbol{B}$ and output matrix $\boldsymbol{C}$ as shown in the equations 1 along with other motor parameters are given.

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{x}(t) &= \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{B}\boldsymbol{u}(t), \\ \boldsymbol{y}(t) &= \boldsymbol{C}\boldsymbol{x}(t)\boldsymbol{D}\boldsymbol{u}(t). \end{aligned} \tag{1}$$

There is no feed through matrix $\boldsymbol{D}$ (D=0) and the system has 4 states (stator currents and rotor fluxes) as well as two inputs (voltages). Two states that are rotor fluxes can not be measured so an observer (Kalman Filter) is to be designed for observing all the states.

A dynamic state reference generator is also provided which produces four sinusoidal waveforms that will be used as reference tracking of the states.

A. Discretization of the system

Continuous time system LTI matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ are calculated by substituting the values of the parameters given in the task using matlab. In order to design a discrete time controller, the continuous time system is discretized by using sampling frequency 10 kHz using the following equations 2. Matrix $\boldsymbol{C}$ remains the same.

$$\begin{aligned} \boldsymbol{A}_d &= \exp^{\boldsymbol{A}T_s}, \\ \boldsymbol{B}_d &= (\boldsymbol{A}_d - \boldsymbol{I})\boldsymbol{A}^{-1}\boldsymbol{B}. \end{aligned} \tag{2}$$

In order to design an observer, model should be fully observable and stable. The given system fullfils both the requirements with full rank observability matrix and stable eigen values in continuous time as well as the discrete time system with the given sampling time.

B. Design of steady state Kalman filter

Now we have a discrete time system which is fully observable and can be represented as the following equation 3.

$$\begin{aligned} \boldsymbol{x}[k+1] &= \boldsymbol{A}_d\boldsymbol{x}[k] + \boldsymbol{B}_d\boldsymbol{u}[k] + \boldsymbol{m}[k], \\ \boldsymbol{y}[k] &= \boldsymbol{C}_d\boldsymbol{x}[k] + \boldsymbol{n}[k]. \end{aligned} \tag{3}$$

Kalman Filter is the optimality counter part of Linear Quadratic Regulator LQR. $\boldsymbol{m}[k]$ is the process noise and represents uncertainty regarding the model inaccuracy and $\boldsymbol{n}[k]$ is the measurement noise. Both $\boldsymbol{m}[k]$ and $\boldsymbol{n}[k]$ are uncorrelated and mean free with a constant covariance $\boldsymbol{M}$ and $\boldsymbol{N}$ respectively as shown in the equation 4. Low $\boldsymbol{M}$ means model is trustworthy. High $\boldsymbol{N}$ means measurement is not trustworthy as sensors as high noise so $\boldsymbol{K}$ correction will be minor. The optimal state estimation objective is to minimize the quadratic cost function.

$$\begin{aligned} \boldsymbol{M} &= \boldsymbol{E}(\boldsymbol{m}[k] + \boldsymbol{m}^{\boldsymbol{T}}[k], \\ \boldsymbol{N} &= \boldsymbol{E}(\boldsymbol{n}[k] + \boldsymbol{n}^{\boldsymbol{T}}[k]. \end{aligned} \tag{4}$$

By using the above knowledge and more by a trail and error approach we can set the values of matrices $\boldsymbol{M}$ and $\boldsymbol{N}$ as

$$\boldsymbol{M} = \begin{bmatrix} 0.0001 & 0 & 0 & 0 \\ 0 & 0.0001 & 0 & 0 \\ 0 & 0 & 0.0001 & 0 \\ 0 & 0 & 0 & 0.0001 \end{bmatrix}$$

$$\boldsymbol{N} = \begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix}$$

To calculate the Kalman gain matrix $\boldsymbol{K}$ we first calculate $\boldsymbol{P}$ matrix (estimation covariance matrix) which is symmetric and positive semi-definite. It is a diagonal matrix with elements containing the estimate variances. Large matrix values indicates high uncertainty.

In kalman filter corrector and predictor model, we perform two steps simultaneosly of prediction and correction. In prediction step, we estimate the covariance matrix $\hat{P}[k+1|k]$ and states $\hat{x}[k+1|k]$ and predict one step ahead using the current inputs and outputs at step [k]. In the correction step, we first determine $K[k+1|k]$ matrix using estimation covariance matrix $\hat{P}[k+1|k]$ and estimated states $\hat{x}[k+1|k]$ and then correct them to get $\hat{P}[k+1|k+1]$ and $\hat{x}[k+1|k+1]$.

Form an implementation point of view Kalman filter is computationally demanding. $\hat{\boldsymbol{P}}[k+1|k]$ and $\hat{\boldsymbol{P}}[k+1|k+1]$ should be updated in each step. Also $\boldsymbol{K}[k+1]$ matrix needs to be updated for each step which invloves a cumbersome inverse calculation.

However, for LTI systems with constant system matrices, we can achieve convergence of Kalman matrix K resulting in

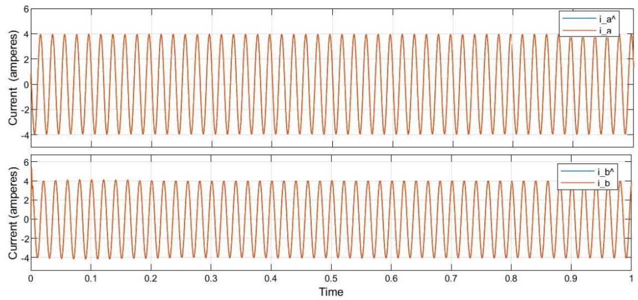

Fig. 1. Estimated vs Actual states $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ (currents)

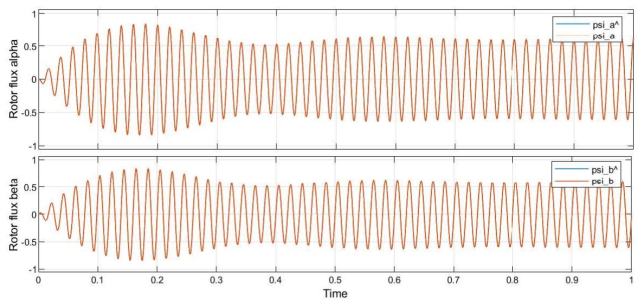

Fig. 2. Estimated vs Actual states $\boldsymbol{x}_3$ and $\boldsymbol{x}_4$ (rotor fluxes)

a steady state K matrix $[k->\infty]=\boldsymbol{K}$. This will make $\hat{\boldsymbol{P}}$ a constant and can be calculated by discrete time Algebraic Riccati equation (DARE) as shown in equations 5.

$$\hat{\boldsymbol{P}} = \boldsymbol{A}\hat{\boldsymbol{P}}\boldsymbol{A^T} - \boldsymbol{A}\hat{\boldsymbol{P}}\boldsymbol{C^T}(\boldsymbol{C}\hat{\boldsymbol{P}}\boldsymbol{C^T})^{-1}\boldsymbol{C}\hat{\boldsymbol{P}}\boldsymbol{A^T} + \boldsymbol{M}. \quad (5)$$

With appropriate initialization $\hat{\boldsymbol{x}}[0|0] = \hat{\boldsymbol{x}}[0])$, the $\boldsymbol{K}$ matrix will be simplified and calculated as equation 6

$$\boldsymbol{K} = \hat{\boldsymbol{P}}\boldsymbol{C^T}(\boldsymbol{C}\hat{\boldsymbol{P}}\boldsymbol{C^T})^{-1}, \quad (6)$$

After solving the above two equations 5 and 6, the K matrix is calculated as

$$\boldsymbol{K} = \begin{bmatrix} 0.064 & 0 \\ 0 & 0.064 \\ 0 & 0.0029 \\ 0.0029 & 0 \end{bmatrix}$$

The results of the estimated states generated from Kalman filter with the above choices of M and N in comparison to the actual states from the plants can be seen in figures 1 and 2. The states generated from the observer are nearly the same as the states of the actual plant. These states will now be fed to the MPC Controller.

C. Design of state-tracking Model Predictive Control (MPC)

Now the next task is to design the state-tracking Model Predictive Control (MPC). Here the reference is dynamic and there is no sensible $\boldsymbol{u}_e$ and the quadratic cost function is adapted as given in the equation 7. It consists of three weighting matrices $\boldsymbol{Q}$, $\boldsymbol{R}$ and $\boldsymbol{S}$. Instead of penalizing the absolute input value $\boldsymbol{u}[k]$, we introduce costs w.r.t. the input change between two sampling steps

$$\begin{aligned} J &= (\boldsymbol{r}[kf] - \boldsymbol{x}[kf]))^T\boldsymbol{S}(\boldsymbol{r}[kf] - \boldsymbol{x}[kf]) \\ &+ \sum(\boldsymbol{r}[k] - \boldsymbol{x}[k]))^T\boldsymbol{Q}(\boldsymbol{r}[k] - \boldsymbol{x}[k]) + (\boldsymbol{u}[k] \\ &- \boldsymbol{u}[k-1])^T\boldsymbol{R}(\boldsymbol{u}[k] - \boldsymbol{u}[k-1]). \end{aligned} \quad (7)$$

Using the condensed input representation, the modified input costs can be rewritten as shown in 8.

$$\begin{aligned} J &= (\tilde{\boldsymbol{R}}_{kf} - \tilde{\boldsymbol{C}}_{kf}\boldsymbol{X}_{kf})^T\boldsymbol{Q}_{kf}(\tilde{\boldsymbol{R}}_{kf} - \tilde{\boldsymbol{C}}_{kf}\boldsymbol{X}_{kf}) + (\boldsymbol{U}_{kf} \\ &-(\boldsymbol{E}\boldsymbol{U}_{old} - \boldsymbol{D}\boldsymbol{U}_{kf}))^T\boldsymbol{R}_{kf}(\boldsymbol{U}_{kf} - (\boldsymbol{E}\boldsymbol{U}_{old} - \boldsymbol{D}\boldsymbol{U}_{kf})) \end{aligned} \quad (8)$$

In the condensed model, we insert the value of $\boldsymbol{X}_{kf}$ in the above cost function as given in 9

$$\boldsymbol{X}_{kf} = \boldsymbol{A}_{kf}\boldsymbol{x}_0 + \boldsymbol{B}_{kf}\boldsymbol{U}_{kf} \quad (9)$$

The overall cost function depending now just on the unknown input can be rewritten as shown in 10. .

$$\begin{aligned} J_u &= 2(\boldsymbol{x}_0^T\boldsymbol{A}_{kf}^T\tilde{\boldsymbol{C}}_{kf}^T\boldsymbol{Q}_{kf}\tilde{\boldsymbol{C}}_{kf}\boldsymbol{B}_{kf} - \tilde{\boldsymbol{R}}_{kf}^T\boldsymbol{Q}_{kf}\tilde{\boldsymbol{C}}_{kf}\boldsymbol{B}_{kf} - \\ &\boldsymbol{U}_{old}^T\boldsymbol{E}^T\boldsymbol{R}_{kf}(\boldsymbol{I}-\boldsymbol{D}))\boldsymbol{U}_{kf} + \boldsymbol{U}_{kf}^T(\boldsymbol{B}_{kf}^T\tilde{\boldsymbol{C}}_{kf}^T\boldsymbol{Q}_{kf}\tilde{\boldsymbol{C}}_{kf} \\ &\boldsymbol{B}_{kf} + (\boldsymbol{I}-\boldsymbol{D})^T\boldsymbol{R}_{kf}(\boldsymbol{I}-\boldsymbol{D}))\boldsymbol{U}_{kf} \end{aligned} \quad (10)$$

The state constraints are represented using a regular polytope with 24 vertices (only current constraints exist, the fluxes remain unconstrained). The input constraints take on the shape of a hexagon. The terminal state constraints do not differ from the general state constraints. Constraints are represented as equation 11

$$\begin{aligned} \boldsymbol{W}_x\boldsymbol{B}_{kf}\boldsymbol{U}_{kf} &\leq \boldsymbol{\omega}_x - \boldsymbol{W}_x\boldsymbol{A}_{kf}\boldsymbol{x}_0, \\ \boldsymbol{W}_u\boldsymbol{U}_{kf} &\leq \boldsymbol{\omega}_u. \end{aligned} \quad (11)$$

To solve the equations 10 and 11, the Quadratic Program (QP) optimization method is employed as given in 12

$$\begin{aligned} \min_z \frac{1}{2}z^T\boldsymbol{H}z + \boldsymbol{f}^Tz, \\ \boldsymbol{G}z \leq \boldsymbol{e} \end{aligned} \quad (12)$$

Now by comparing the cost function and the Quadratic program, the cost terms are found to be equation 13,

$$\begin{aligned} \boldsymbol{H} &= 2(\boldsymbol{B}_{kf}^T\tilde{\boldsymbol{C}}_{kf}^T\boldsymbol{Q}_{kf}\tilde{\boldsymbol{C}}_{kf}\boldsymbol{B}_{kf} + (\boldsymbol{I}-\boldsymbol{D})^T\boldsymbol{R}_{kf}(\boldsymbol{I}-\boldsymbol{D})), \\ \boldsymbol{f}^T &= 2(\boldsymbol{x}_0^T\boldsymbol{A}_{kf}^T\tilde{\boldsymbol{C}}_{kf}^T\boldsymbol{Q}_{kf}\tilde{\boldsymbol{C}}_{kf}\boldsymbol{B}_{kf} - \tilde{\boldsymbol{R}}_{kf}^T\boldsymbol{Q}_{kf}\tilde{\boldsymbol{C}}_{kf}\boldsymbol{B}_{kf} - \\ &\boldsymbol{U}_{old}^T\boldsymbol{E}^T\boldsymbol{R}_{kf}(\boldsymbol{I}-\boldsymbol{D})). \end{aligned} \quad (13)$$

while the constraints terms are represented as the following

$$\boldsymbol{G} = \begin{bmatrix} \boldsymbol{W}_x\boldsymbol{B}_{kf} \\ \boldsymbol{W}_u \end{bmatrix}$$

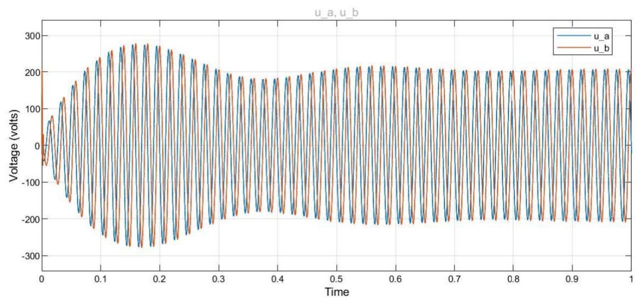

Fig. 3. Plant input $\boldsymbol{u}_a$ and $\boldsymbol{u}_b$ (volts)

$$\boldsymbol{e} = \begin{bmatrix} \boldsymbol{\omega}_x - \boldsymbol{W}_x \boldsymbol{A}_{kf} \boldsymbol{x}_0 \\ \boldsymbol{\omega}_u \end{bmatrix}$$

The QP optimization result is $z^* = \boldsymbol{U}_{kf}$ and, therefore, we do not need to perform any additional coordinate transform before we apply the first m elements from z^* as the control input.

The weighting matrices $\boldsymbol{Q}, \boldsymbol{R}$ and $\boldsymbol{S}$ for the controller design correspond to a fast transient response with an acceptable tolerance for the measurement noise. The weighting matrix $\boldsymbol{Q}$ penalizes the state deviations from reference and the weighting matrix $\boldsymbol{R}$ penalizes the change in control effort. The weighting matrix $\boldsymbol{S}$ penalizes a possible remaining state deviation at time t_f. We will take $\boldsymbol{S}$ = $\boldsymbol{Q}$ and after trial and error of different combination of values of $\boldsymbol{Q}$ and $\boldsymbol{R}$ matrices, following values give fastest transient response and least input spike. The results are compiled in a tabular form in I.

$$\boldsymbol{Q} = \begin{bmatrix} 100 & 0 & 0 & 0 \\ 0 & 150 & 0 & 0 \\ 0 & 0 & 100 & 0 \\ 0 & 0 & 0 & 100 \end{bmatrix}$$

$$\boldsymbol{R} = \begin{bmatrix} 0.001 & 0 \\ 0 & 0.05 \end{bmatrix}$$

D. Findings and Analysis

TABLE I
INPUT OPTIMIZATION TABLE

Pred Horizon	Pk-Volt	Trkg $\boldsymbol{x}_1$	Trkg $\boldsymbol{x}_2$	Trkg $\boldsymbol{x}_3$	Trkg $\boldsymbol{x}_4$
kf=1	290	0s	-	-	-
kf=2	275	0s	0.4s	0.7s	0.7s
kf=3	271	0s	0s	0.5s	0.5s

The objective of the whole task is to find the minimum prediction horizon that produces a near optimal performance of state tracking. This ensures reduced real-time computational constraints on the processor.

The following three figures corresponds to the performance of the MPC. The choice of k_f and sampling time is very important keeping in view the physical constraints of the real world hardware. We would like to have a small sampling time and well as large k_f, but doing so will burden our model.

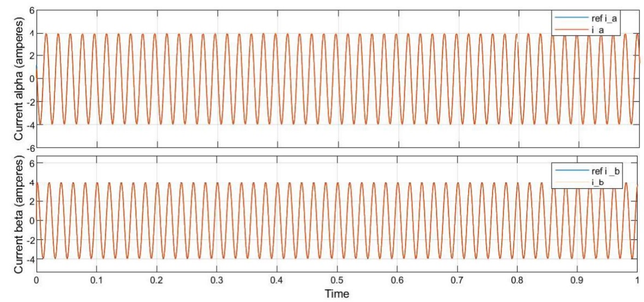

Fig. 4. Reference tracking of states $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ (currents)

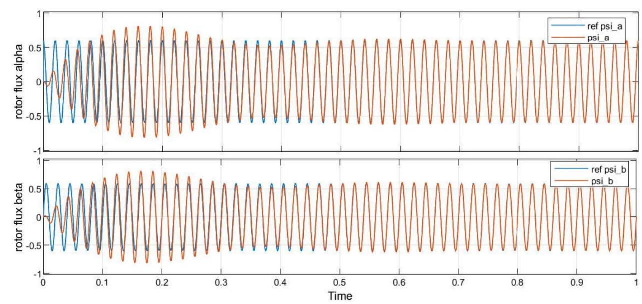

Fig. 5. Reference tracking of states $\boldsymbol{x}_3$ and $\boldsymbol{x}_4$ (rotor fluxes)

Since sampling time given is already small, we are limited for the choice of k_f. The prediction horizon k_f is chosen to be 3. This is because the response of minimum input and state tracking is the best and did not significantly improve further. Increasing the k_f will increased the range of initial feasible state and makes the controller more farsighted but we have to choose for optimality. Therefore, k_f = 3 is the optimal value with regards to speed and computation. Figure 3 shows the input to the plant. A peak voltage of 272 V and an initial spike of 235 V is observed. The comparison of different prediction horizons and input and state tracking times can be found the the I

Figure 4 and 5 shows the reference signal from the reference generator and all the states of the plant. The graph depicts that stator currents are tracked very fast and there is no overshoot or oscillations in it. However, the rotor flux tracking is slow (0.5s). As per the demonstrated figures, it can be concluded that the performance of the IM controller is fairly good. This is because even though two states are not measured directly, it is able to estimate the states with minimum error and track the sinusoidal reference. The only limitation is the tracking speed of rotor flux reference.