# A model predictive control approach for electrical motor drive

Qingyi Ren(5684803), Liuyi Zhu(5808383)

***Abstract*—In this project, the regulation MPC, Output MPC with disturbance rejection and adaptive MPC with time-varying reference are explored and implemented to drive model to achieve the desired angular speed of the electrical motor. The stability of equilibrium at certain control input is analysed and proved to be asymptotic. Finally the numerical simulations are illustrated and the properties of different MPC algorithms are discussed.**

## I. Introduction

The project is mainly focusing on controlling the speed of motor by using the voltage of the drive. The construction of the dynamic model of the drive is partly based on the essay [1]. The dynamics of electrical motor drive are nonlinear and could be linearized at equilibrium. The system with the reference frame $(d, q)$ has four states and two control inputs which can be coupled as MIMO system. Three different MPC algorithms are implemented in different system conditions and scenarios. Then the control input and state constraints are implemented and the system properties e.g. stability considering system constraints is further be analysed.

## II. Drive Model

In this project, a focus is put on the speed and current control of a PMSM drive. Looking at the electrical subsystem of PMSM drive, with the reference frame $(d, q)$, the electrical dynamics of PMSM drive is then obtained as:

$$\begin{aligned} \frac{\mathrm{d}i_d}{\mathrm{d}t} &= \frac{1}{L_d}(u_d - Ri_d + \omega_{me}L_q i_q) \\ \frac{\mathrm{d}i_q}{\mathrm{d}t} &= \frac{1}{L_q}(u_q - Ri_q - \omega_{me}L_d i_d - \omega_{me}\Lambda_{mg}) \end{aligned} \tag{1}$$

Then the mechanical dynamics is shown as:

$$\frac{\mathrm{d}\omega_{me}}{\mathrm{d}t} = \frac{p}{J}(k_t i_q - \frac{B}{p}\omega_{me} - \tau_L) \tag{2}$$

where $\omega_{me} = p\omega_m$ and $\omega_{me}$ depicts the electromechanical speed, $\Lambda_{mg}$ represents the PM flux linkage, $\omega_m$ is the mechanical speed, $\tau_L$ is the disturbance torque, $J$ and $B$ are the moment of inertia the viscous coefficient of the load and the motor constant is $k_t = 3p\Lambda_{mg}/2$.

The electrical and mechanical dynamics could compose the PMSM drive system, which consists of a set of nonlinear equations. Firstly, for convenience of measurement and coupling terms, the state is defended as:

$$x = \begin{bmatrix} i_d & i_q & \omega_{me} \end{bmatrix}^T$$

Qingyi Ren and Liuyi Zhu are master students at TU Delft, The Netherlands. E-mail addresses: {Q.REN-2, L.Zhu-6}@student.tudelft.nl.

where $i_d$ and $i_q$ represent currents components in the d-axis and p-axis respectively, $\omega_{me}$ is the angular speed and the measured disturbance denoted as $\widehat{\omega_{me} i_q}$. The input $u$ represents the components of voltage vector in the d-axis and p-axis respectively:

$$u = \begin{bmatrix} u_d & u_q \end{bmatrix}^T$$

Neglecting the $\omega_{me} i_d$ term in the equation (2), and linearizing at the equilibrium $[0, 0, 0]^T$, the state-space model of drive model is shown as:

$$\begin{aligned} A &= \begin{bmatrix} \frac{\partial f_1(x,u)}{\partial x_1} & \frac{\partial f_1(x,u)}{\partial x_2} & \frac{\partial f_1(x,u)}{\partial x_3} \\ \frac{\partial f_2(x,u)}{\partial x_1} & \frac{\partial f_2(x,u)}{\partial x_2} & \frac{\partial f_2(x,u)}{\partial x_3} \\ \frac{\partial f_3(x,u)}{\partial x_1} & \frac{\partial f_3(x,u)}{\partial x_2} & \frac{\partial f_3(x,u)}{\partial x_3} \end{bmatrix} \\ &= \left.\begin{bmatrix} -\frac{R}{L_d} & \frac{L_q x_3}{L_d} & \frac{L_q x_2}{L_d} \\ -\frac{L_d x_3}{L_q} & -\frac{R}{L_d} & -\frac{L_d x_1}{L_q} - \frac{\Lambda_{mg}}{L_q} \\ 0 & \frac{Pk_t}{J} & -\frac{B}{J} \end{bmatrix}\right|_{x=x_e} \\ &= \begin{bmatrix} -\frac{R}{L_d} & 0 & 0 \\ 0 & -\frac{R}{L_d} & -\frac{\Lambda_{mg}}{L_q} \\ 0 & \frac{Pk_t}{J} & -\frac{B}{J} \end{bmatrix} \end{aligned} \tag{3}$$

$$B = \begin{bmatrix} \frac{\partial f_1(x,u)}{\partial u_1} & \frac{\partial f_1(x,u)}{\partial u_2} \\ \frac{\partial f_2(x,u)}{\partial u_1} & \frac{\partial f_2(x,u)}{\partial u_2} \\ \frac{\partial f_3(x,u)}{\partial u_1} & \frac{\partial f_3(x,u)}{\partial u_2} \end{bmatrix} = \begin{bmatrix} \frac{1}{L_d} & 0 \\ 0 & \frac{1}{L_q} \\ 0 & 0 \end{bmatrix} \tag{4}$$

$$\dot{x} = \begin{bmatrix} -\frac{R}{L_d} & 0 & 0 \\ 0 & -\frac{R}{L_d} & -\frac{\Lambda_{mg}}{L_q} \\ 0 & \frac{Pk_t}{J} & -\frac{B}{J} \end{bmatrix} x + \begin{bmatrix} \frac{1}{L_d} & 0 \\ 0 & \frac{1}{L_q} \\ 0 & 0 \end{bmatrix} u \tag{5}$$

By employing the Forward Euler discretisation method, the continuous-time dynamic model (5) could be further transformed to the discrete-time dynamic model:

$$\begin{aligned} x(k+1) = &\begin{bmatrix} 1 - T\frac{R}{L_d} & 0 & 0 \\ 0 & 1 - T\frac{R}{L_q} & -T\frac{\Lambda_{mg}}{L_q} \\ 0 & T\frac{pk_t}{J} & 1 - T\frac{B}{J} \end{bmatrix} x(k) + \\ &\begin{bmatrix} \frac{T}{L_d} & 0 \\ 0 & \frac{T}{L_q} \\ 0 & 0 \end{bmatrix} u(k) = Ax(k) + Bu(k) \end{aligned} \tag{6}$$

where $x(k)$ represents discrete-time state at time step $k$,

$$x(k) = [i_d(kT), i_q(kT), \omega_{me}(kT)]^T$$

, $T$ represents the sample time and $u(k) = \begin{bmatrix} u_d(kT) \\ u_q(kT) \end{bmatrix}$ as control inputs.

In this project, the parameters mentioned before are set as $L_d = 1.2 \times 10^{-3}$ H, $L_q = 0.8 \times 10^{-3}$ H, $R = 0.15\ \Omega$, $p = 4$, $J = 0.02$, $B = 0.01$ and $\Lambda_{mg} = 0.15$. The sampling period is set as $T = 0.01s$.

## III. Model predictive control design

In this section, three different MPC tragedies are applied to the drive motor system to achieve the desired speed and enforce both the current and the voltage limits at the same time.

1) Regulation MPC is applied to stabilize the system around the equilibrium point.
2) Output MPC is used for constant reference tracking with disturbance rejection.
3) Adaptive MPC is utilized for the time-varying reference

It is worth mentioning that the three MPC algorithms make use of different structures of cost functions which will be elaborated thoroughly in the following subsections.

### A. Regulation MPC

Regulation MPC method is capable of dynamically modeling a system and predicting system's future dynamics. This control method can achieve optimal control and increase system stability while meeting input and output constraints.

Regulation MPC control method enables the motor system to reach an steady state from different starting points. In this part, the equilibrium of system lies at origin $[0, 0, 0]^T$. The linearization is done around the equilibrium 0 in equation (6). In reality $x_k \to 0$ means the state of the motor system, including the current and angular speed of motor, can reach 0 as quickly as possible by applying the optimal input voltage under different operating points.

The state equation of the system is shown as followed:

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) \\ y(k) &= Cx(k) \end{aligned} \tag{7}$$

where $C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$. Before designing the MPC controller, it is necessary to analyze the observability and controllability of the system to ensure that the system can be well-modeled and controlled. We construct the controllability matrix and obvservability matrix in MATLAB and check the rank of the matrix. The rank of both matrices is equal to the number of system states. Therefore, the system is controllable and observable.

It is also necessary to check whether the system is stable. By obtaining the eigenvalues of $A$ matrix in MATLAB, it was found that all the eigenvalues are within the unit circle, indicating that the system is stable.

The regulation MPC problem can be represented as follows:

- Objective function:

$$\begin{aligned} J(x_0, u) = \sum_{k=0}^{N-1} l(x(k), u(k)) + V_f(x(N)) = \frac{1}{2}(\mathbf{x}^T Q \mathbf{x} \\ + \mathbf{u}^T R \mathbf{u} + x(N)^T P x(N)) \end{aligned} \tag{8}$$

- Constraints and parameters:
  1) state equation:$x_{k+1} = Ax_k + Bu_k$, where $x_k \in \mathbb{R}^n$ is the state vector at time step $k$, and $u_k \in \mathbb{R}^m$ is the control input vector at time step $k$.
  2) known initial state:$x_0$
  3) state constraints:$x_k \in X$,where $X \subseteq \mathbb{R}^n$.
  4) input constraints:$u_k \in U$,where $U \subseteq \mathbb{R}^m$.
  5) prediction horizon:$N$.

The objective of regulation MPC problem is to find an optimal control sequence $u_0, u_1, \ldots, u_{N-1}$ that minimizes the cost function $J$ while satisfying the dynamic equation and input and state constraints.

The cost function consists of stage cost and terminal cost, where the state sequence $\mathbf{x} = [x(0), x(1), \ldots, x(N-1)]^T$ and control input sequence $\mathbf{u} = [u(0), u(1), \ldots, u(N-1)]^T$.

$Q$ and $R$ are positive definite symmetric matrices. In our system, the third state is the motor angular speed, which is generally one order of magnitude larger than the current. To balance the differences in magnitude, we adjusted the weight for angular speed to be smaller, in order to prevent excessive influence on the system from changes in rotational speed. Therefore, we determine the $Q$ and $R$ matrix as followed:

$$Q = \alpha \cdot diag\begin{pmatrix} 1 & 1 & 0.1 \end{pmatrix} \quad R = diag\begin{pmatrix} 1 & 1 \end{pmatrix}$$

where $\alpha$ is adjustable parameter. In section V of the report, simulation results of different parameter choices will be compared and analyzed. $P$ is the solution of discrete algebraic Riccati equation (DARE) for unconstrained infinite-horizon LQR problem, which can contribute to the stability proof in Section IV.

In this motor system, the absolute value of the motor current should be limited to within $1A$, and the absolute value of the motor speed should be limited to within $30r/s$, while speed with negative values represents the motor running in reverse. The state constraints can be expressed as followed:

$$\mathbb{X} := \{x_k \in \mathbb{R}^3 \mid |x_1| \le 1, |x_2| \le 1, |x_3| \le 30\} \tag{9}$$

As for the input voltage of the system, its absolute value is limited to within 3V. The input constraints can be expressed as followed:

$$\mathbb{U} := \{u_k \in \mathbb{R}^2 \mid |u_1| \le 3, |u_2| \le 3\} \tag{10}$$

The constraint sets of state and the input $x \in X$ , $u \in U$ are closed and bounded and contain the origin, which satisfy assumption 2.3 in stability analysis in Section IV.
There are also multiple choices for the initial values $x_0$, sampling period $T$ and prediction horizon $N$ of the system, which will be analyzed in section V respectively.

### B. Output MPC

Compared with regulation MPC, output MPC only uses the system's output signal as input to the controller, without using the system's state variables. By predicting and optimizing the output signal, output MPC can achieve stable and optimal control of the system's output.

In reality, disturbances exist in the state update and output measurements of system, and it aims to control the motor to maintain a constant angular speed. Therefore, the disturbance is added to the system and an output MPC controller is designed to track a constant output reference while rejecting disturbance.

After added the disturbance, the state equation of system is shown as followed:

$$x_{k+1} = Ax_k + Bu_k + B_d d \tag{11}$$

$$y_k = Cx_k + D_d d \tag{12}$$

where $d = \begin{bmatrix} 0.1 & 1 \end{bmatrix}^T$ is a constant disturbance and

$$B_d = \begin{bmatrix} 1 & 0.5 \\ 0.4 & 0 \\ 0 & 1 \end{bmatrix} \quad C_d = \begin{bmatrix} 0.5 & 0 \\ 0 & 2 \end{bmatrix} \tag{13}$$

The updated state equation of system can also be expressed as an augmented system:

$$\begin{bmatrix} x_{k+1} \\ d_{k+1} \end{bmatrix} = \begin{bmatrix} A & B_d \\ 0 & I \end{bmatrix} \begin{bmatrix} x_k \\ d_k \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u_k \tag{14}$$

$$y_k = \begin{bmatrix} C & C_d \end{bmatrix} \begin{bmatrix} x_k \\ d_k \end{bmatrix} \tag{15}$$

Assume $x_e = \begin{bmatrix} x & d \end{bmatrix}^T$,$A_e = \begin{bmatrix} A & B_d \\ 0 & I \end{bmatrix}$, $B_e = \begin{bmatrix} B \\ 0 \end{bmatrix}$,$C_e = \begin{bmatrix} C & C_d \end{bmatrix}$, the system's state equation can be expressed in standard form:

$$x_e^+ = Ax_e + Bu \tag{16}$$

$$y = C_e x_e \tag{17}$$

The output $y$ is measured, but the state $x$ and the disturbance $d$ are unknown. Based on that, a Luenberger observer is created to estimate the state and disturbance:

$$\hat{x}_e^+ = A_e \hat{x}_e + B_e u + L(y - C_e \hat{x}_e) \tag{18}$$

where $L$ is the observer gain, which makes $(A_e - LC_e)$ stable. We can calculate $L$ in MATLAB by placing the poles of the system.

Since the original system is observable, the augmented system is also observable and

$$rank \begin{bmatrix} I - A & -B_d \\ C & C_d \end{bmatrix} = n + n_d = 5 \tag{19}$$

After verifying the necessary conditions mentioned before, the output MPC controller is designed as:

- Objective function:

$$\begin{aligned} V_N(x_{e0}, y_r, u) &= \sum_{k=0}^{N_e-1} l(x_e(k) - x_r, u(k) - u_r) \\ &+ V_f(x_e(N_e) - x_r) = \frac{1}{2}((\mathbf{x}_e - \mathbf{x}_r)^T Q_e (\mathbf{x}_e - \mathbf{x}_r) \\ &+ (\mathbf{u} - \mathbf{u}_r)^T R_e (\mathbf{u} - \mathbf{u}_r) + x_e(N)^T P_e x_e(N_e)) \end{aligned} \tag{20}$$

- Constraints and parameters:
  1) state equation:$x_e^+ = A_e x_e + B_e u$, where $x_e \in \mathbb{R}^{n+n_d}$ is the state vector, and $u \in \mathbb{R}^m$ is the control input vector.
  2) known initial state:$x_{e0}$
  3) constant output reference:$y_r$
  4) state constraints:$x_e \in X$,where $X_e \subseteq \mathbb{R}^{n+n_d}$.
  5) input constraints:$u \in U$,where $U_e \subseteq \mathbb{R}^m$.
  6) prediction horizon:$N_e$.

where $\mathbf{x}_r$ and $\mathbf{u}_r$ represent the target state and control input vectors including the time steps $i = 0, 1, \ldots N_e - 1$. By combining with the state space function, the optimal target selection could be written as:

$$(x_r, u_r)(\hat{d}, y_r) \in \begin{cases} \min_{x_r, u_r} & ||u_r||^2 \\ s.t. & \begin{bmatrix} I - A & -B \\ C & 0 \end{bmatrix} \begin{bmatrix} x_r \\ u_r \end{bmatrix} = \begin{bmatrix} B_d \hat{d} \\ y_r - C_d \hat{d} \end{bmatrix} \\ & x_r \in X_e \\ & u_r \in U_e \end{cases} \tag{21}$$

The $OTS$ will be solved in every iteration loop based on the estimate of disturbance. The workflow of output feedback offset-free MPC is applied to measure the current output and uses the output to update state and disturbance estimates. With disturbance estimate we can use $OTS$ to select optimal target and then solve the optimal control problem. Finally, the optimal input is applied to update state and measure the output in next time step.

The cost function for output MPC is constructed by building the tracking error between the real output and reference. As the system has been augmented, the weight matrices also need to be augmented and the weighting for disturbance state should be 0. Therefore, we determine $Q_e$, $R_e$ and $P_e$ matrix as followed:

$$Q_e = \alpha \cdot diag \begin{pmatrix} 1 & 1 & 0.1 & 0 & 0 \end{pmatrix}$$

$$R_e = diag \begin{pmatrix} 1 & 1 \end{pmatrix}$$

$$P_e = diag \begin{pmatrix} P & 0 & 0 \end{pmatrix}$$

The state constraints and input constraints of output MPC are the same as in regulation MPC:

$$\mathbb{X}_e := \{x_e \in \mathbb{R}^5 \mid |x_1| \leq 1, |x_2| \leq 1, |x_3| \leq 30\} \tag{22}$$

$$\mathbb{U}_e := \{u \in \mathbb{R}^2 \mid |u_1| \leq 3, |u_2| \leq 3\} \tag{23}$$

from the structure of constraint sets of state and input, it could be inferred that these sets are compact and contain the origin.

### C. Adaptive MPC

In order to capture the time-varying reference, Adaptive MPC is used to increase the accuracy of prediction. It could adapt the parameters of the prediction model at each iteration while keeping the fixed model structure. The implementation of Adaptive MPC is carried out by using MPC toolbox in MATLAB [2] and it will be elaborated in section A.

At each iteration, the plant model is updated by adaptive MPC controller, but will not change during its current iteration which lasts for the length of prediction horizon.

## IV. Asymptotic stability

In our system, the equilibrium point 0 only means when given zero control input the state is equal to zero (including the angular speed equal to 0), in real life it is expected that the speed achieves a certain positive value or satisfies a certain function with time which has most values bigger than zero, thus it is not that meaningful to analyse the stability of equilibrium 0 of original system.

Based on that, the linearization around a positive state and control input is needed. It is assumed that the given input is $u = [2, 2]^T$, the equilibrium $x_e$ is calculated according to the equations (1) and (2). Defining the state $[i_d, i_q, \omega_{me}]^T = [x_1, x_2, x_3]^T$, the equilibrium could be calculated as:

$$\begin{aligned} \dot{x_1} &= f_1(x,u) = (u_1 - Rx_1 + L_q x_2 x_3)/L_d = 0 \\ \dot{x_2} &= f_2(x,u) = (u_2 - Rx_2 - L_d x_1 x_3 - \Lambda_{mg} x_3)/L_q = 0 \\ \dot{x_3} &= f_3(x,u) = Pk_t x_2/J - Bx_3/J = 0 \end{aligned} \tag{24}$$

By applying $u_1 = 2$ and $u_2 = 2$, the equilibrium at this input could be computed as $x_e = [0.35, -0.39, 18]^T$. The linearised state space model could be computed in the same procedure in Section III.

In order to transform the equilibrium $x_e$ to the origin, the transform is used [3]. It can be defined that $z = x - x_e$, then the state-space model could be represented as:

$$\dot{x} = \dot{z + x_e} = \dot{z} = A(z + x_e) + Bu = Az + \underbrace{Ax_e}_{d} + Bu \tag{25}$$

The constant term $d = Ax_e$ could be then taken as the disturbance added to the state-space model, by constructing the augmented input [4], the new state-space model could be represented as:

$$\dot{z} = Az + \begin{bmatrix} B & F \end{bmatrix} \begin{bmatrix} u \\ d \end{bmatrix} = Az + \bar{B}\bar{u} \tag{26}$$

where $\bar{B} = \begin{bmatrix} B & F \end{bmatrix}$ and $\bar{u} = \begin{bmatrix} u \\ d \end{bmatrix}$ where $F = I \in \mathbb{R}^{3\times 3}$, $d = x_e$. By using this method, now the new state-space model has the origin as its equilibrium.

### A. Linearized stability Analysis

From the former analysis, the dynamics of motor drive is linearized around the $x_e$, having the linear form of system (26). Firstly, the controllability of the linearized system should be examined. The Kalman matrix is constructed as $K = \begin{bmatrix} B & AB & A^2B \end{bmatrix}$, the Kalman matrix has the full row rank, which makes sure that the system is controllable. The state cost could be written in the quadratic form:

$$l(x,u) = \sum_{k=0}^{N-1} l(x(k), u(k)) = \frac{1}{2}(\mathbf{x}^T Q \mathbf{x} + \mathbf{u}^T R \mathbf{u}) \tag{27}$$

Choosing $Q = \alpha \cdot diag(1, 1, 0.1)$ and $R = diag(1, 1)$ as $Q, R \succ 0$, both of them are positive definite, the stage cost has the lower bound as:

$$l(x,u) \geq \frac{1}{2}\mathbf{x}^T Q \mathbf{x} \geq \frac{1}{2}\lambda_{min}(Q)\left\|x\right\|^2 = \alpha_1(|x|) \tag{28}$$

where $\lambda_{min}(Q)$ means the smallest eigenvalue of matrix $Q$. The terminal cost could be represented as $V_f(x) = \frac{1}{2}x^T P x$ where $P$ is the solution to the Discrete Algebraic Riccati Equation (DARE):

$$0 \prec P = A_K^T P A_K + Q_K \tag{29}$$

where $A_K = A + BK$, $Q_K = Q + K^T R K$ and $K$ is the optimal unconstrained LQR gain $K = -(B^T P B + R)^{-1} B^T P A^T$. As the matrix $P$ is positive definite, the upper bound of terminal cost could be obtained $V_f(x) = \frac{1}{2}x^T P x \leq \frac{1}{2}\lambda_{max}(P)\left\|x\right\|^2 = \alpha_f(|x|)$, where $\lambda_{max}(P)$ means the largest eigenvalue of matrix $P$. In the closed-loop system, the MPC state feedback law is applied using the optimal unconstrained LQR gain $u = K_N(x) = Kx$, the closed loop system could be further represented as $x^+ = f(x,u) = (A + BK)x = A_K x$. In order to infer the asymptotically stable of the origin, the following assumptions in the book [5] should be made first:

**Assumption 2.2**: Continuity of system and cost: the functions $l(x,u)$, $V_f(x)$ and $x^+ = f(x,u)$ are continuous and $l(0,0) = 0$, $V_f(0) = 0$ and $f(0,0) = 0$.

In our case, the $l(x,u)$, $V_f(x)$ and $x^+ = f(x,u)$ are linear or in the quadratic form, which could ensure the continuity of system. At the same time the requirements of $l(0,0) = 0$, $V_f(0) = 0$ and $f(0,0) = 0$ are satisfied.

**Assumption 2.3**: The constraint sets of state and the input $x \in X$ , $u \in U$ are closed and bounded and contain the origin. In section III, the properties of constraints sets have already been verified to safitsfy the Assumption 2.3.

**Assumption 2.14**: Basic stability assumption:

(1) For all $x \in X_f$, there exists a u, such that $(x,u) \in Z$ satisfying

$$f(x,u) \in X_f$$

$$V_f(f(x,u)) - V_f(x) \leq -l(x,u)$$

(2) There exists $\kappa_\infty$ functions $\alpha_1(\cdot)$ and $\alpha_f(\cdot)$ satisfying

$$l(x,u) \geq \alpha_1(|x|), \forall x \in X_N,\ \forall u \text{ such that } (x,u) \in Z$$
$$V_f(x) \leq \alpha_f(|x|), \forall x \in X_f$$

where $X_N$ is the region of attraction.

In our case, the control invariant set will be estimated in the section IV-B, based on control invariant set $X_f$, the Assumption 2.14(1) could be examined properly.

For Assumption 2.14(3), the analysis of Assumption 2.2 has already confirmed that there exists $\kappa_\infty$ functions $\alpha_1(\cdot)$ and $\alpha_f(\cdot)$. More specifically, $\alpha_1(|x|)$ and $\alpha_f(|x|)$ could be written as $\alpha_1(r)$ and $\alpha_f(r)$ functions which are strictly increasing with $r \in [0, \infty)$ and $\alpha_1(0) = 0$, $\alpha_f(0) = 0$. Furthermore, when $r \to \infty$, $\alpha_1 \to \infty$ and $\alpha_f \to \infty$. (according to the definition of $\kappa_\infty$ functions)

### B. Estimating $X_f$

Finding the set $X_f$ is equal to obtaining the maximum invariant constraint admissible set and the set $X_N$ is identical to find the region of attraction. The initial conditions in $X_N$ could reach the terminal set $X_f$ in $N$ steps.

The algorithm 1 shown in Appendix B presents the procedure of estimating the inequality of set $X_f$, $Hx \leq h$. The set

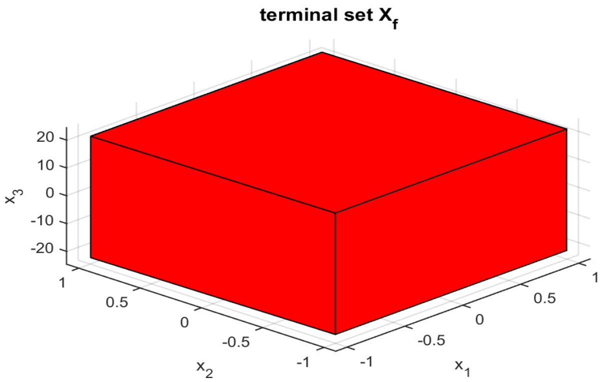


Fig. 1: Terminal set $X_f$

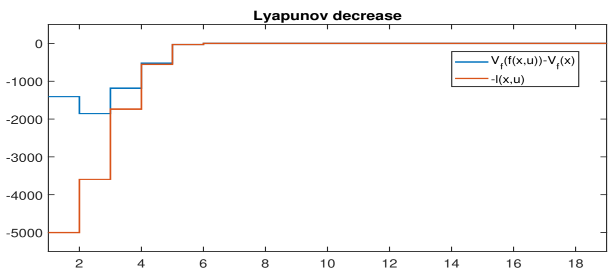


Fig. 2: The lyapunov decrease starting from one initial state in set $X_N$ but outside $X_f$

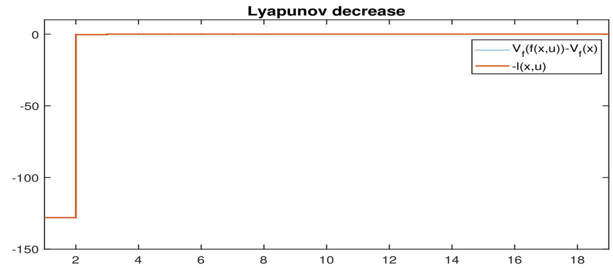


Fig. 3: The lyapunov decrease starting from one initial state in set $X_f$

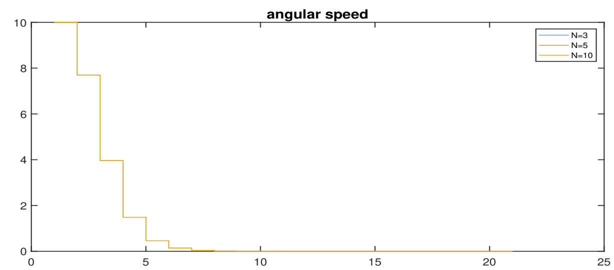


Fig. 4: Output for different $N$

$X_f$ is estimated which contains 6 inequalities having the form as:

$$X_f = \left\{ x \in R^3 | Hx \le h, H \in R^{6\times 3}, h \in R^{6\times 1} \right\}$$

In order to make the estimated $X_f$ satisfy the states and input constraints, the state-to-state mapping ($x = Ix$) and state-to-input ($u = Kx$) mapping are made by constructing $K_{aug} = [K; I]$. This could directly calculate the state and input admissible invariant sets instead of the output admissible set. By using the Multi-Parametric Toolbox 3 to build LTI system and then employ the function **system.invariantSet()**, the terminal set $X_f$ was illustrated in Figure (1), which had the structure of parallelepiped.

The algorithm 1 in Appendix B shows how to estimate $X_N$ according to $X_f$. By checking the feasibility of MPC problem with terminal constraint $x(N) \in X_f$ starting from a relatively large initial state set, the combination of all initial states which make the MPC problem feasible could be taken as $X_N$.

### C. Verifying Stability Assumption

Firstly the Assumption 2.14(a) is to be confirmed. It states that for all $x \in X_f$ there exists a $u$ which ensures $(x, u) \in Z$ and the Lyapunov decrease is satisfied:

$$V_f(f(x,u)) - V_f(x) \le -l(x,u) \tag{30}$$

We choose two representative initial points, the first point is $[0.1; 0.1; 10]$ which settles in the set $X_N$ but outside the set $X_f$ and the other point is $[0.04; 0.04; 1.6]$ which is just in the set $X_f$. In Figure (2), the value $V_f(f(x,u)) - V_f(x)$ is larger than the value $-l(x,u)$ in the first 5 steps, but at the step $N$, the state reached the terminal set $X_f$ and the Lyapunov decrease is satisfied, that is, $V_f(f(x,u)) - V_f(x) = -l(x,u)$ after the $N$ steps. For comparison, the Figure (3) with the initial state in set $X_f$, as it is already the in the terminal set, the Lyapunov decrease is always satisfied, with $V_f(f(x,u)) - V_f(x) = -l(x,u)$. It can be inferred that for all initial states belonging to $X_N$ but outside $X_f$, the Lyapunov decrease will hold after $N$ steps (equal to prediction horizon).for all initial states belonging to the set $X_f$ the Lyapunov decrease will hold.

The other part of Assumption 2.14(a) is that for all $x \in X_f$, there exists a $u((x,u) \in Z)$ satisfying

$$f(x,u) \in X_f \quad \forall x \in X_f$$

By considering that $X_f$ is invariant set, all states start within $X_f$ will remain in $X_f$. $X_f$ is already be computed by Algorithm 1 in Appendix B which satisfies inequality $Hx \le h$.

## V. Numerical simulations

In this section, the simulative results are shown and the influence of parameters on the MPC controllers are discussed. Based on different scenarios, the comparison of different MPC controllers is made to analyse the application scenes for MPC controllers.

### A. Regulation MPC with different predictive horizon

In this part, we choose different predictive horizon to simulate the system. The choice for predictive horizon is 3, 5 and 10 respectively. The motor angular speed and system input are shown in Figures 4,5 and 6. Other system parameters are $\alpha = 1$ and $x_0 = \begin{bmatrix} 0.1 & 0.1 & 10 \end{bmatrix}^T$.

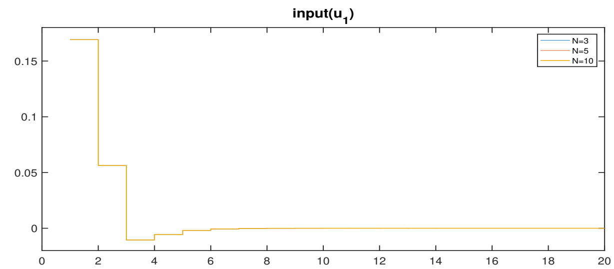


Fig. 5: $u_1$ for different $N$

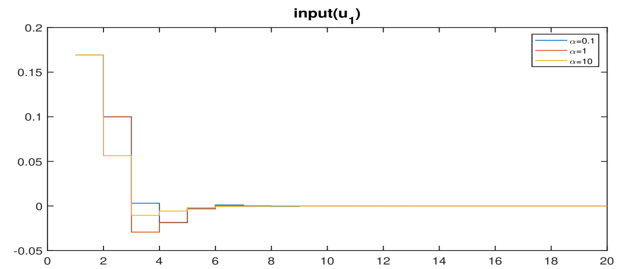


Fig. 8: $u_1$ for different weighting matrix

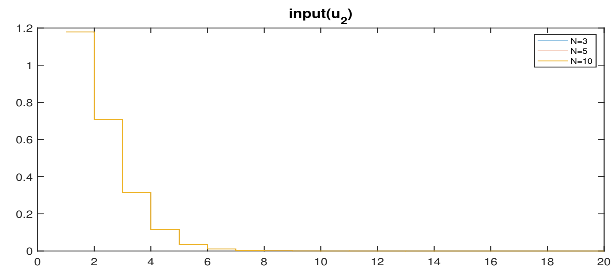


Fig. 6: $u_2$ for different $N$

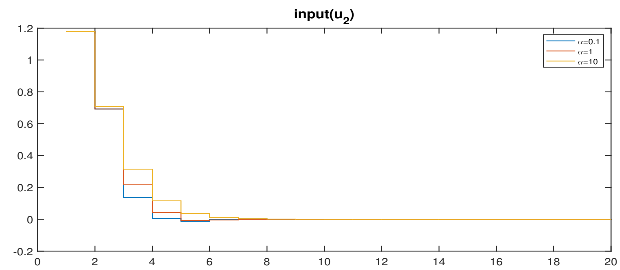


Fig. 9: $u_2$ for different weighting matrix

Increasing the predictive horizon allows the controller to optimize over a longer future horizon, which may lead to better control performance. However, larger predictive horizon could also increase the computational burden of the controller and may make the controller more sensitive to the system's external interference and internal parameter variation.

From the graphs, it is clear that the simulation results for different predictive horizons do not show significant differences. The reason for this phenomenon may be that the motor system has a small inertia and then causes a fast response speed, which allows the controller to respond quickly and achieve good control performance within a short prediction horizon.

### B. Regulation MPC with different cost weights

In this part, the influence of different weighting matrices on the controller will be analyzed and compared. We have chosen values of $\alpha$ as 0.1, 1 and 10 respectively and plotted the simulation results. The motor speed and controlled input are shown in Figures 7,8 and 9. Other simulation parameters are $N = 5$ and $x_0 = \begin{bmatrix}0.1 & 0.1 & 10\end{bmatrix}^T$.

When $\alpha$ has smaller values, it means that the weight for the system state is also smaller and the weight for the system input is larger. In order to minimize the objective function, the system will accelerate its response and make the controlled input return to 0 as soon as possible.

From these graphs, it can be observed that in this motor system, the smaller the value of $\alpha$, the faster the system responses and the more likely it is to have an overshoot. If $\alpha$ is larger, the system response is smoother but slower. To balance the system's response speed and stability, we choose $\alpha$ to be 1.

### C. Regulation MPC with different starting points

We select different initial points to simulate the system. The choice for starting points is $\begin{bmatrix}0.1 & 0.1 & 10\end{bmatrix}^T$, $\begin{bmatrix}0.3 & 0.3 & 15\end{bmatrix}^T$ and $\begin{bmatrix}0.5 & 0.5 & 20\end{bmatrix}^T$ respectively. The motor angular speed and system input are shown in Figure 10,11 and 12. Other system parameters are set as $\alpha = 1$ and $N = 5$.

From the figure, we can see that the further the initial state of the system is from the equilibrium state, the larger controlled input of the system is needed, and the more steps it takes to return to the equilibrium state. If we choose the initial state outside the region of attraction, it may lead to system instability.

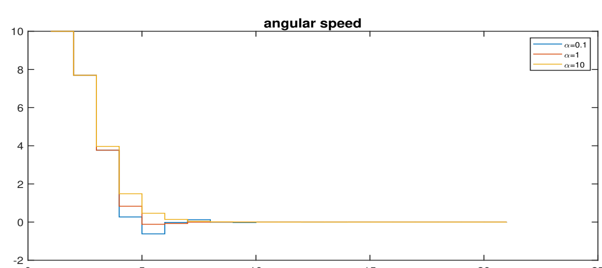


Fig. 7: Output for different weighting matrix

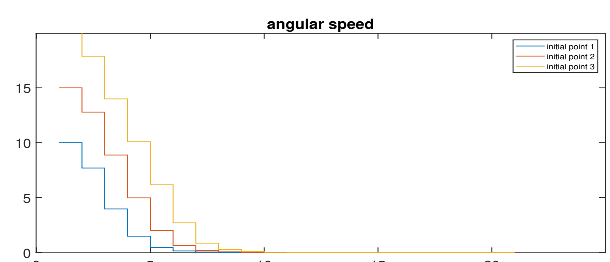


Fig. 10: Output for different initial points

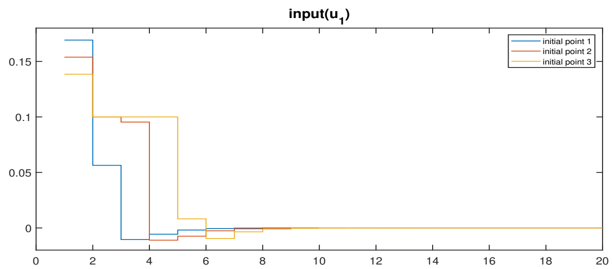


Fig. 11: $u_1$ for different initial points

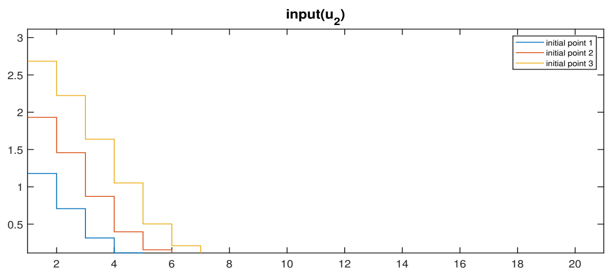


Fig. 12: $u_2$ for different initial points

## D. Offset-free MPC(output feedback)

The aim of output MPC is to track the output reference while rejecting the disturbance. Based on the simulation in previous part, we set the system parameters as followed:

$$\alpha_e = 1, N_e = 5, y_r = \begin{bmatrix} 0.4 & 20 \end{bmatrix}^T \tag{31}$$

Figure 13 shows the tracking error between true speed and reference. The tracking error gradually approaches zero over time, which indicates that the system successfully tracked the reference value and rejected disturbances. The control input of system is shown in Figure 14 and 15.

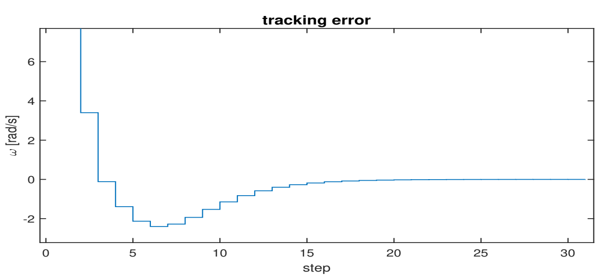


Fig. 13: Tracking error

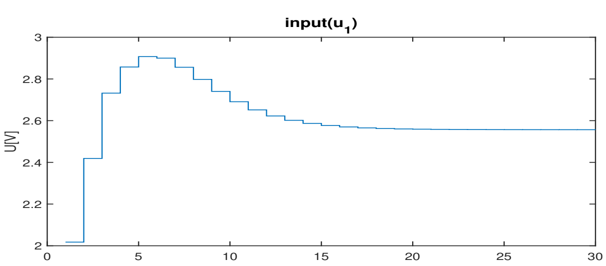


Fig. 14: $u_1$ (output MPC)

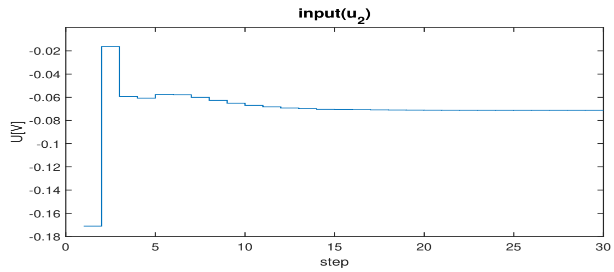


Fig. 15: $u_2$ (output MPC)

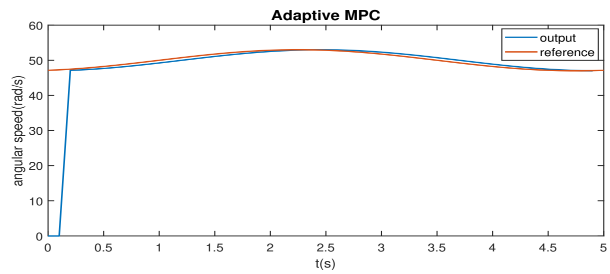


Fig. 16: The reference of angular speed and the output $y_3$

## E. Adaptive MPC

The Adaptive MPC is built by using MATLAB Model Predictive Control Toolbox, the reference of the angular speed $\omega_{me}$ is set to be time-varying **sine** function, the output $y_3$ could not only show a fast response which reaches $47.467$ at $t = 0.2$s, but also fit the reference precisely with the **sine** wave after $t = 0.2$s shown in figure (16). It can show that Adaptive MPC has a good adaptability to the change of system, in our case the change condition is the reference. More details about Adaptive MPC programming part are shown in A.

# VI. Conclusion

The motor system is a non-linear system which can be linearised around the equilibrium. In this assignment, we try to design different MPC controllers to regulate the angular speed of the electrical model and analyse the stability around the equilibrium at certain control input.

By applying various MPC control methods (regulation MPC, output MPC and adaptive MPC), we have gained a deeper understanding of the principles of MPC control, the impact of different control parameters on the system, and stability theory in MPC control.

# Appendix

## A. Adaptive MPC controller

The Adaptive MPC is built by using Model Predictive Control Toolbox in Matlab [2]. This can create the MPC object with several structure variables, e.g. ManipulatedVariables which represent the states of system, constraints of outputs and states and weights which define the tuning weights of outputs and states. Furthermore it can tune object properties of MPC object, for example, to change the prediction horizon. In Adaptive MPC design, the sample time is set to be 0.1s.

```
%constraints
con.umin=0;
con.umax=100;
con.imax=3;
con.imin=0;
%State space model
sysa=ss(sys.A,sys.B,sys.C,sys.D,para.T);
%Build MPC controller obejctive
mpcobj = mpc(minreal(sysa), para.T);
mpcobj.PredictionHorizon = 6;% Prediction horizon
mpcobj.ControlHorizon = 3;%Control horizon
%State and output weights
mpcobj.Weights.OutputVariables = [1 10000];
mpcobj.Weights.ManipulatedVariablesRate = [10 10];
%State and output constraints
mpcobj.OutputVariables(2).Min = con.imin;
mpcobj.OutputVariables(2).Max = con.imax;
mpcobj.OutputVariables(1).Min = con.imin;
mpcobj.OutputVariables(1).Max = con.imax;
mpcobj.ManipulatedVariables(1).Min = con.umin;
mpcobj.ManipulatedVariables(1).Max = con.umax;
mpcobj.ManipulatedVariables(2).Min = con.umin;
mpcobj.ManipulatedVariables(2).Max = con.umax;

Nsima = 50;
t = linspace(0, Nsima*para.T, Nsima+1);
r = [3*ones(size(t));3*sin(2*pi*0.2*(t-1))+50*ones(
    size(t))];
mpcobj.Model.Nominal = struct('y', [0;0;0], 'u',
    [0;0]);
nominal=mpcobj.Model.Nominal;
xmpc = mpcstate(mpcobj);
x = xmpc.Plant;
YY = []; RR = []; UU = []; XX = [];

for k = 1:Nsima
    XX = [XX;x']; % store plant state
    y = sys.C*x; % calculate plant output
    YY = [YY;y']; % store plant output
    RR = [RR;r(:,k)']; % store reference
    u = mpcmoveAdaptive(mpcobj,xmpc,sysa,nominal,y,r
    (:,k)); % calculate optimal mpc move
    UU = [UU;u']; % store plant input
    x = sys.A*x+sys.B*u; % update plant state
    % is the last line necessary since x=xmpc.Plant
    gets updated anyway
end
```

### *B. Algorithms to estimate $X_f$ and $X_N$*

Algorithms 1 and 2 are listed below to estimate Sets $X_f$ and $X_N$.

**Algorithm 1**: Estimating Set $X_f$

**Result**: Obtain the inequality of $Hx \leq h$ representing $X_f$
**Initialization**:
$A_k$:=$A - BK$
$K$:= LQR optimal gain
$K_{aug}$:= $[K; I]$
set $k$:=0
**Iteration**:
**For all** $i = 1, 2, \cdots, s$

$$x_i^* := \begin{cases} \underset{x}{\operatorname{argmax}} f_i\left(K_{\text{aug}} A_K^{k+1} x\right) \\ \text{s.t. } f_j\left(K A^t x\right) \leq 0 \quad \forall j \in \{1,2,\ldots,s\}, \\ \qquad \forall t \in \{0,1,\ldots,k\} \end{cases} Z$$

**end**
**if**

$$f_i\left(K_{\text{aug}} A_K^{k+1} x\right) \leq 0 \forall i \in \{0,1,\ldots,s\}$$

**then**

$$X_f := \{x \in \mathbb{R}^n \mid f_i\left(K_{\text{aug}} A_K^t x\right) \leq 0 \forall j \in \{1,2,\ldots,s\} \\ \forall t \in \{0,1,\ldots,k\}\}$$

**else**

set $k$:=$k+1$ and continue

**end**

**Algorithm 2**: Estimating Terminal Set $X_N$

**Result**: Obtain an approximation of $X_N$ which denoted as $X_N^{'}$
**Initialization**:
$P_N(x_0, u)$
$X_f$:=$\{x \in R^n | Hx \leq h\}$
$X_{ICs}$ set of ICs to consider
$X_N^{'}$:=$\{\}$
**Iteration**:
**For all** $x_0 \in X_{ICs}$
**if** MPC problem is feasible:
s.t. $x(N) \in X_f$
$u \in U$, $x \in X$
**then**
$X_N^{'} = x_0 \cup X_N^{'}$
**end**
**end**

## REFERENCES


[1] S. Bolognani, S. Bolognani, L. Peretti, and M. Zigliotto, "Design and implementation of model predictive control for electrical motor drives," *IEEE Transactions on Industrial Electronics*, vol. 56, no. 6, pp. 1925–1936, 2009.

[2] M. M. Alberto Bemporad and N. L. Ricker, *Model Predictive Control Toolbox: For Use with MATLAB*. The Mathworks, 2008.

[3] H. Huang, "Feedback systems," https://www.cds.caltech.edu/~murray/amwiki/index.php/FAQ:_If_every_equilibrium_point_can_be_transformed_to_the_origin_and_then_analyzed_using_a_Lyapunov_function,_how_can_a_system_have_both_stable_and_unstable_equilibrium_points%3F.

[4] Paul, "Mathworks," https://nl.mathworks.com/matlabcentral/answers/723093-state-space-with-disturbance.

[5] J. Rawlings and D. Mayne, *Model Predictive Control: Theory and Design*. Nob Hill Publishing, 2008.