BİR HIZLI MODEL ÖNGÖRÜLÜ KONTROL METODU UYGULAMASI

Application of a Fast Model Predictive Control Method

Barış DÜNDAR & Doç. Dr. Özge CİHANBEĞENDİ
Department of Electrical and Electronics Engineering, Dokuz Eylul University, Izmir, Turkey.
ozge.sahin@deu.edu.tr

ÖZET: Kontrol teorisinde, Lineer Zamanla Değişmeyen (LTI) dinamik sistemlerin kararlılığı ve kontrolü için Model Öngörülü Kontrol (MPC) algoritmaları, kısıtlar üzerinde öngörülen plant model kullanılarak bir Quadratik Programı (QP) online olarak çözmektedir. Klasik MPC algoritmalarında, hesaplama Kayan Horizon zamanı üzerinde yapımaktadır. Bir MPC problemi bir QP'ye dönüştürüldüğünde, QP nümerik yöntemler kullanılarak çözülür. Yavaş ve büyük ölçekli sistemlerde, MPC algoritmaları başarılıdır ve onun başarısı hızlı sistemlere uygulanabilir; ancak MPC algoritmalarının hızlı sistemlere uygulanabilmesinde bazı hesaplama problemleri (yavaş operasyon ve bir optimal çözüme yakınsama gibi) bulunmaktadır. Literatürde, bu sorunları giderebilmek için online ve offline MPC yöntemleri geliştilmektedir. Bu makale, bu yöntemlerin hızlı sistemlere daha etkin uygulanabilmesi için önerilen yeni bir yöntemin uygulamasını sunmaktadır. Uygulamada, basitleştirilmiş Quansar helikopterin (plant) açı dinamiklerinin kontrol ve kararlılığı analiz edilmiştir. Analizde, bir MPC algoritmasının plant'a göre daha hızlı hesaplama yapması amaçlanmıştır. Bu amaç için, plant'ın zaman ölçekli çözümleri bir MPC algoritmasının plant model'i olarak kullanılmıştır. Böylece, hızlı hesaplama yoluyla plant'ın kontrolü ve kullanılan kontrol algoritmaları geliştirilebilir.

Anahtar Kelimeler: Model Öngörülü Kontrol, Hızlı Hesaplama, Lineer Dinamik Sistemler.

ABSTRACT: In control theory, Model Predictive Control (MPC) algorithms for stability and control of Linear Time-Invariant (LTI) dynamical systems solve a Quadratic Program (QP) as online using prediction plant model on constraints. In classical MPC algorithms, the computation is done over Receding Horizon time. Once an MPC problem is converted to a QP, it is solved using numerical methods. In slow and high – scaled systems, MPC algorithms

are successful and its success can be applied for fast systems but there exists some computation issues (slow operation and convergence to an optimal solution) to be able to apply MPC algorithms to fast systems. In literature, online and offline MPC methods have been improving to be able to compansate these issues. This article presents an application of a proposed novel method to be able to apply these methods to fast systems more effective. In the application, the control and stability of the angles dynamics of the simplified Quansar helicopter (plant) are analyzed. In the analysis, an MPC algorithm aims faster computation according to the plant. For this aim, the time scaled solutions of the plant are used as plant model of any MPC algorithm. Thus, the control of plant and the used control algorithms can be improved by fast computation.

Keywords: Model Predictive Control, Fast Computation, Linear Dynamical Systems.

1. INTRODUCTION

Model Predictive Control (MPC), which is used for control and stability of dynamical systems, is an online optimal control strategy and consists of parts of plant model or prediction model, cost function, constraints, optimization, and Receding Horizon Principle (RHP). In MPC literature, all the researches focus on the five parts of MPC [1-2].

In this article, the aim is to present the application of a method used in order to make the applicability of the existing fast MPC algorithms [3-22] to fast and parallel operating microprocessors. For this aim, any MPC algorithm is sped up by scaling of solutions of plant on time domain. The scaled solutions of plant are used as plant model of MPC algorithms. This method is called as upsampled MPC (UMPC) and UMPC algorithm also creates free times between plant and any MPC algorithm at sampling intervals of plant. Thus, the MPC algorithms can be improved using free times for more effective control strategies (See Fig.1) [24].

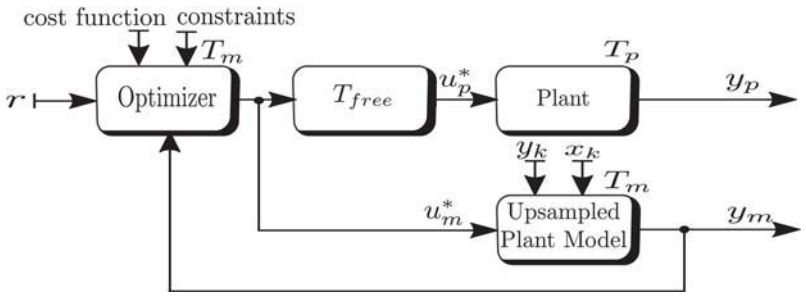


Figure 1: UMPC Block Diagram

2. UMPC METHOD

Consider the deterministic, Multi-Input Multi-Output (MIMO) and controllable Linear Time Invariant (LTI) plant is given by the following state equation [23]

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) \\ x(0) &= x_0, \quad \forall \ t \geq t_0 \end{aligned} \tag{1}$$

where state variable $x(t) \in \mathbb{R}^n$, initial state variable $x_0 \in \mathbb{R}^n$, input or control variable $u(t) \in \mathbb{R}^r$, output variable $y(t) \in \mathbb{R}^s$, constant matrices $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times r}$, $C \in \mathbb{R}^{n\times r}$, and continuous time variable $t \in \mathbb{R}^+$.

Consider the following state equations are the plant model, which is used for any MPC algorithm [24]

$$\begin{aligned} \dot{x}(t) &= M(Ax(t) + Bu(Mt)) \\ y(t) &= Cx(t) \\ x(t_0) &= x_0, \quad \forall \ t \geq t_0 \end{aligned} \tag{2}$$

where $M > 1$ is an integer time scaling factor.

A time scaling relation can be written between the solution $x(t) \stackrel{\Delta}{=} x_p(t)$ of the plant (1) and the solution $x(t) \stackrel{\Delta}{=} x_m(t)$ of the plant model (2)

$$x_m(t) = x_p(Mt) \tag{3}$$

The following relation can be written between the discrete-time solution $x_p(k)$ of the plant (1) at control sampling interval T_p and the discrete-time solution $x_m(k)$ of the plant model (2) at control sampling interval $T_m = T_p/M$ [24]

$$x_m(kT_m) = \{x_p(kT_p) |\ \text{wait } T_{free} \quad \forall\ kT_m\} \tag{4}$$

where the free time at each time instant k is

$$\begin{aligned} T_{free} &= T_p - T_m \\ &= T_m(M-1) \end{aligned} \tag{5}$$

The discrete-time plant model with sampling interval T_m of the plant model (2) is called as the Upsampled Plant Model (UPM) and it can be used for the control and the stability of the plant (1).

The prediction model for the UPM of plant model (2) with the sampling interval $T_m = T_p/M$ is

$$\begin{aligned} x(k+i+1) &= A_{md}x(k+i) + B_{md}u_m(k+i) \\ y(k+i) &= Cx(k+i) \\ x(k) &= x_k, \quad \forall \ k, i \geq 0 \end{aligned} \tag{6}$$

where the matrices $A_{md} \in \mathbb{R}^{n\times n}$ and $B_{md} \in \mathbb{R}^{n\times r}$ at T_m, x_k is the estimated initial state value from the plant (1) and then discretized by $T_m = T_p/M$ at time instant k.

In MPC algorithms, control laws are computed over state prediction N_p and control prediction N_c horizons as $i = 0,1,\dots,N_c,\dots,N_p$ at each discrete time instant $k, k+1, k+2, \dots$ Assume $N_p = N_c = N$. The state and control prediction vectors at k are given by the following forms

$$\mathbf{x}(k) = \Theta x_k + \Phi \mathbf{u_m}(k) \tag{7}$$

where

$$\begin{aligned} &\mathbf{x}(k) = [x(k+1) \quad x(k+2) \quad\quad x(k+N)]^T \in \mathbb{R}^{Nn} \\ &\mathbf{u_m}(k) = [u_m(k) \quad u_m(k+1) \quad\quad u_m(k+N-1)]^T, \\ &\mathbf{u_m}(k) \in \mathbb{R}^{Nr} \end{aligned} \tag{8}$$

$$\Theta = \begin{bmatrix} A_{md} & A_{md}^2 & \dots & A_{md}^N \end{bmatrix}^T \in \mathbb{R}^{Nn \times n} \tag{9}$$

$$\Phi = [\Phi_1 \quad \Phi_2 \quad \dots \quad \Phi_N]^T = \begin{bmatrix} B_{md} & 0 & \dots & 0 \\ A_{md}B_{md} & B_{md} & & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A_{md}^{N-1}B_{md} & A_{md}^{N-2}B_{md} & & B_{md} \end{bmatrix} \in \mathbb{R}^{Nn \times Nr} \tag{10}$$

The UPM is subject to the linear state and input rate prediction constraints with $T_m = T_p/M$

$$\begin{aligned} &u_{min} \le u_m(k+i) \le u_{max} \\ &x_{min} \le x(k+i) \le x_{max} \end{aligned} \tag{11}$$

Any MPC algorithm, which operates at T_p, is desired to operate based on the UPM equation (6) and its optimization sampling interval T_{opt} to be $T_{opt} = T_m$ (i.e. UMPC). Thus, UMPC algorithm can be written as [24-25]

$$\begin{aligned} J_k^*(x_k) &= \min_{\Delta \mathbf{u_m}(k)} J_k(x_k, \mathbf{u_m}(k)) \\ &= \sum_{i=0}^{N-1} [x^T(k+i) q_w x(k+i) + u_m^T(k+i) r_w u_m(k+i) \\ &\quad + x^T(k+N) p_N x(k+N)] \\ &\text{subject to} \\ &u_{min} \le u_m(k+i) \le u_{max} \qquad i = 0,1,\dots,N-1 \quad \forall\ k \\ &x_{min} \le x(k+i) \le x_{max} \qquad i = 1,\dots,N-1 \quad \forall\ k \\ &u_{min} \le K(A_{md} + B_{md}K)^i x(k+N) \le u_{max} \qquad i = 0,1,\dots,\epsilon \quad \forall\ k \\ &x_{min} \le (A_{md} + B_{md}K)^i x(k+N) \le x_{max} \qquad i = 0,1,\dots,\epsilon \quad \forall\ k \\ &x(k) = x_k \qquad \text{at } k \end{aligned} \tag{12}$$

where $J_k(x_k, \mathbf{u_m}(k))$ is the objective function, $J_k^*(x_k)$ is the optimal cost value, the weighted matrices q_w, (or $q_w = C^T C$), r_w and p_N are positive definite. The state feedback gain K is computed on the infinite horizon and unconstrained form (Linear Quadratic Optimal Control Problem) of the UMPC algorithm (20) and the minimum horizon ϵ of the terminal constraints are determined by the defined linear programs. Note that UMPC algorithm (12) is MPC algorithm for the plant model and optimization with T_p [25].

UMPC problem is converted to an optimization problem. This optimization problem is a quadratic program (QP) , which has a global minimum optimal control law $\mathbf{u_m^*}(k)$ in Karush-Kuhn-Tucker (KKT) conditions. The optimal control laws is solved by using numeric methods (such as active set and interior point methods) [6].

If physical constraint conditions are convenient, (such as ADC and DAC etc.), for a microproccessor or integrated circuit operate with the sampling interval $T_{\mu P}$, M can be selected as the following proportion [24]

$$M = \frac{T_p}{T_{\mu P}} \qquad \text{for integer} \qquad M > 1 \tag{13}$$

Thus, the relationship between the control laws of MPC with T_p and UMPC with T_m algorithms for any k can be written as

$$\mathbf{u_m^*}(kT_m) = \{\mathbf{u_p^*}(kT_p) |\ \text{wait}\ T_{free}\ \ \forall\ kT_m\} \tag{14}$$

where

$$\begin{aligned} T_{free} &= T_p - T_m \\ &= T_m(M-1) \end{aligned} \tag{15}$$

and $\mathbf{u_p^*}(k)$ is the control law vector of MPC with T_p (i.e. classical MPC).

The knowledge of fast control law vector $\mathbf{u_m^*}(kT_m)$ should be saved in a register up to the free time T_{free} and then implemented to the plant at kT_p until next plant sampling interval $(k+1)T_p$ at each time instant k. Thus, $\mathbf{u_m^*}(kT_m)$ can be used instead of $\mathbf{u_p^*}(kT_m)$. Consequently, we achieved free time T_{free} using UMPC algorithm instead of MPC algorithm (See Fig.2).

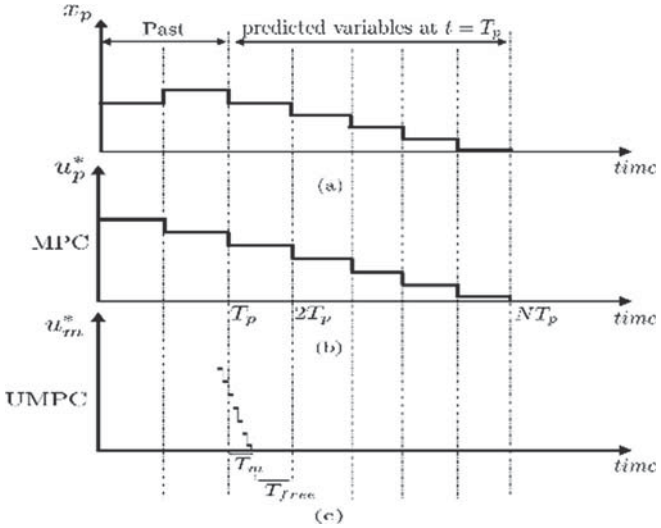


Figure 2: The Plant Control using MPC and UMPC [24]

a. The Predicted Plant State at T_p (k = 1)
b. The Predicted Input at T_p (k = 1) using MPC
c. The Predicted Input at T_m (k = 1) using UMPC

UMPC Algorithm [24-25]

1. Measure the state x_k at kT_m
2. Compute $\mathbf{u}_\mathbf{m}^*(k)$ at kT_m
3. If $\mathbf{u}_\mathbf{m}^*(k) = 0$ Then 'Problem Infeasible' Stop
4. Wait or Improve MPC Algorithm in kT_{free}
5. Implement the first element of $\mathbf{u}_\mathbf{m}^*(k)$ to the plant at kT_p
6. Wait for the new sampling time $(k + 1)T_p$, Go to 1

3. MPC AND UMPC FOR THE ANGLE DYNAMICS OF THE SIMPLIFIED QUANSAR HELICOPTER

Consider the simplified Quansar helicopter dynamics (plant) shown in Fig.3 [26].

$$\begin{aligned}
&\ddot{\theta}_e(t) = u_1(t) + u_2(t)\\
&\ddot{\theta}_r(t) = -\sin(\theta_p(t))\\
&\ddot{\theta}_p(t) = u_1(t) - u_2(t)\\
&\theta_e(0) = \theta_r(0) = \theta_p(0) = 0 \quad \text{rad.}
\end{aligned} \tag{16}$$

with the constraints

$$\begin{aligned}
&-5 \le u_1(t) \le 5 \quad \text{V.}\\
&-5 \le u_2(t) \le 5 \quad \text{V.}\\
&-0.5 \le \theta_e(t) \le 0.6 \quad \text{rad.}\\
&-0.8 \le \theta_r(t) \le 0.9 \quad \text{rad.}\\
&-1 \le \theta_p(t) \le 1 \quad \text{rad.}
\end{aligned} \tag{17}$$

where u_1 and u_2 are the input or control voltages and θ_e, θ_r, θ_p are the elevator, rudder and pitch angles of the plant, respectively. The initial conditions of the angles are zero radian.

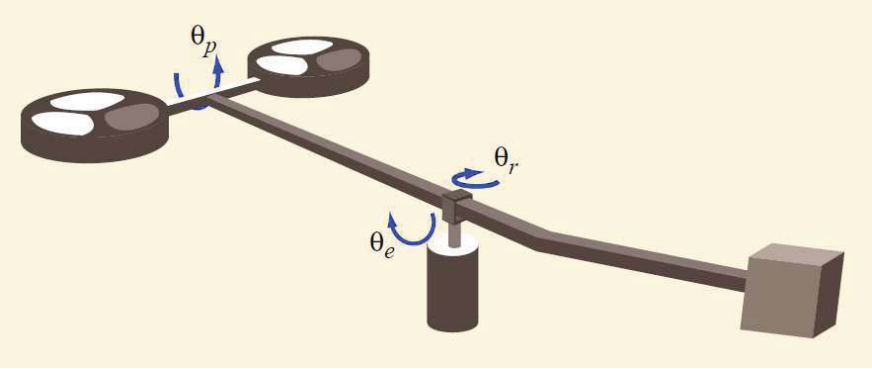


Figure 3: The Angle Dynamics of the Simplified Quansar Helicopter [26]
Using Matlab, the linearized state equations of the plant (16) are obtained

$$\dot{x}(t) = \begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \\ \dot{x}_3(t) \\ \dot{x}_4(t) \\ \dot{x}_5(t) \\ \dot{x}_6(t) \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \\ x_4(t) \\ x_5(t) \\ x_6(t) \end{bmatrix} + \begin{bmatrix} 1 & 1 \\ 0 & 0 \\ 1 & -1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} u_1(t) \\ u_2(t) \end{bmatrix} \tag{18}$$

$$y(t) = \begin{bmatrix} y_1(t) \\ y_2(t) \\ y_3(t) \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \\ x_4(t) \\ x_5(t) \\ x_6(t) \end{bmatrix} \tag{19}$$

$$x(0) = 0, \quad \forall \ t \geq 0$$

about the equilibrium point of the system.

For $M = 100$, the state equations of the plant model are

$$\dot{x}(t) = \begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \\ \dot{x}_3(t) \\ \dot{x}_4(t) \\ \dot{x}_5(t) \\ \dot{x}_6(t) \end{bmatrix} = 100 \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \\ x_4(t) \\ x_5(t) \\ x_6(t) \end{bmatrix} + 100 \begin{bmatrix} 1 & 1 \\ 0 & 0 \\ 1 & -1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} u_1(100t) \\ u_2(100t) \end{bmatrix} \tag{20}$$

$$y(t) = \begin{bmatrix} y_1(t) \\ y_2(t) \\ y_3(t) \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \\ x_4(t) \\ x_5(t) \\ x_6(t) \end{bmatrix}$$

$$x(0) = 0, \quad \forall \ t \geq 0$$

(21)

where the states x_1, x_3, x_5 are the elevator, rudder and pitch angles.

The discrete-time state equations of the linearized plant (18-19) with $T_p = 0.5sec.$ are

$$x(k+1) = \begin{bmatrix} x_1(k+1) \\ x_2(k+1) \\ x_3(k+1) \\ x_4(k+1) \\ x_5(k+1) \\ x_6(k+1) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -0.5 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0.5 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0.5 & -0.125 & 0 & 1 & 0 \\ 0 & 0 & 0.5 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \\ x_4(k) \\ x_5(k) \\ x_6(k) \end{bmatrix} + \begin{bmatrix} 0.5 & 0.5 \\ -0.125 & 0.125 \\ 0.5 & -0.5 \\ 0.125 & 0.125 \\ -0.02083 & 0.02083 \\ 0.125 & -0.125 \end{bmatrix} \begin{bmatrix} u_1(k) \\ u_2(k) \end{bmatrix} \tag{22}$$

$$\begin{aligned} y(k) &= \begin{bmatrix} y_1(k) \\ y_2(k) \\ y_3(k) \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \\ x_4(k) \\ x_5(k) \\ x_6(k) \end{bmatrix} \\ x(0) &= 0, \quad \forall \ k \geq 0 \end{aligned} \tag{23}$$

Note that

$$u_p(k) \triangleq [u_1(k) \quad u_2(k)]^T \tag{24}$$

$$[x_1(k) \quad x_3(k) \quad x_5(k)]^T \triangleq [x_{1p}(k) \quad x_{2p}(k) \quad x_{3p}(k)]^T \tag{25}$$

for the plant (22-23). The dynamics (22-23) are used as the plant model of the classical MPC algorithms.

The discrete-time state equations of the plant model (20-21) with $T_m = 0.005sec.$ are

$$x(k+1)=\begin{bmatrix}x_1(k+1)\\x_2(k+1)\\x_3(k+1)\\x_4(k+1)\\x_5(k+1)\\x_6(k+1)\end{bmatrix}=\begin{bmatrix}1 & 0 & 0 & 0 & 0 & 0\\0 & 1 & -0.5 & 0 & 0 & 0\\0 & 0 & 1 & 0 & 0 & 0\\0.5 & 0 & 0 & 1 & 0 & 0\\0 & 0.5 & -0.125 & 0 & 1 & 0\\0 & 0 & 0.5 & 0 & 0 & 1\end{bmatrix}\begin{bmatrix}x_1(k)\\x_2(k)\\x_3(k)\\x_4(k)\\x_5(k)\\x_6(k)\end{bmatrix}$$
$$+\begin{bmatrix}0.5 & 0.5\\-0.125 & 0.125\\0.5 & -0.5\\0.125 & 0.125\\-0.02083 & 0.02083\\0.125 & -0.125\end{bmatrix}\begin{bmatrix}u_1(k)\\u_2(k)\end{bmatrix} \tag{26}$$

$$y(k)=\begin{bmatrix}y_1(k)\\y_2(k)\\y_3(k)\end{bmatrix}=\begin{bmatrix}0 & 0 & 0 & 1 & 0 & 0\\0 & 0 & 0 & 0 & 1 & 0\\0 & 0 & 0 & 0 & 0 & 1\end{bmatrix}\begin{bmatrix}x_1(k)\\x_2(k)\\x_3(k)\\x_4(k)\\x_5(k)\\x_6(k)\end{bmatrix}$$

$$x(0)=0,\qquad \forall\ k\geq 0 \tag{27}$$

Note that

$$u_{\mathrm{m}}(\mathrm{k}) \triangleq [\mathrm{u}_1(\mathrm{k}) \quad \mathrm{u}_2(\mathrm{k})]^{\mathrm{T}} \tag{28}$$

$$[\mathrm{x}_1(\mathrm{k}) \quad \mathrm{x}_3(\mathrm{k}) \quad \mathrm{x}_5(\mathrm{k})]^{\mathrm{T}} \triangleq [\mathrm{x}_{1\mathrm{m}}(\mathrm{k}) \quad \mathrm{x}_{2\mathrm{m}}(\mathrm{k}) \quad \mathrm{x}_{3\mathrm{m}}(\mathrm{k})]^{\mathrm{T}} \tag{29}$$

for the plant model (26-27).

These dynamics are used as the plant model of the UMPC algorithms, which is M times faster than the sampling interval of the plant (22-23) and the classical MPC algorithms.

For the horizon $\mathrm{N}=10$, the matrices $\mathrm{q_w}=10\mathrm{I}$, $\mathrm{r_w}=10\mathrm{I}$, the scaling factor $\mathrm{M}=100$, the elevator, rudder and pitch reference values $0.35,-0.75$, and 0 rad., respectively, the optimal control law vectors $\mathrm{u_p^*(k)}$ and $\mathrm{u_m^*(k)}$ of the MPC and the UMPC are shown in Fig.4. Thus, we have free time $\mathrm{T_{free}}=0.495$ sec. to be able to implement the control law $\mathrm{u_m^*(k)}$ to plant at each control sampling interval $\mathrm{T_p}=0.5$ sec

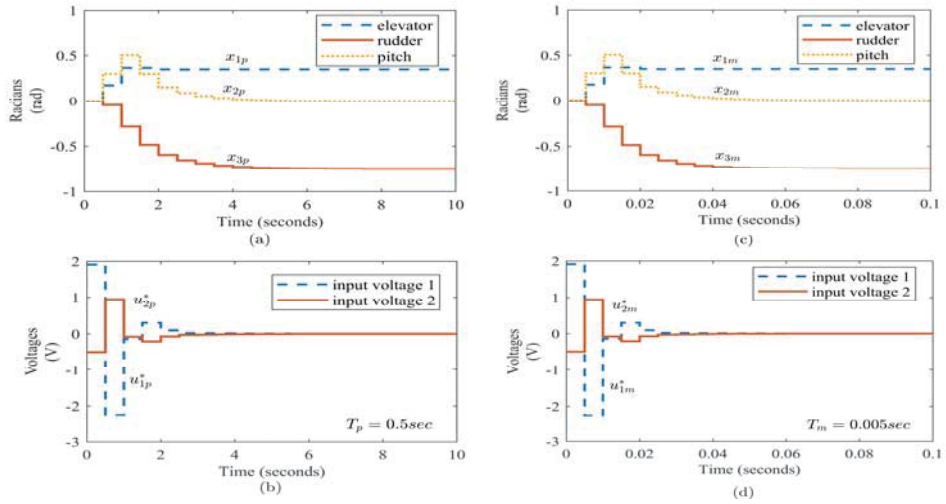


Figure 4: The control of the Quansar helicopter using MPC or UMPC for M = 100

(a) The angles using MPC
(b) The MPC control laws
(c) The angles using UMPC
(d) The UMPC control laws

5. CONCLUSIONS

In this article, the angle dynamics of the simplified Quansar helicopter is controlled using UMPC algorithm and obtained free time at each plant sampling interval. In free times,

• possible inconvenient flight conditions can be predicted,

• plant errors can be compansated and new flights conditions adapted to plant model,

• different control strategies can be implemented such as changing prediction horizon, parallel operationg plant models.

6. REFERENCES

[1] Garcia, C.E., Prett, D.M., Morari M.: 'Model Predictive Control: Theory and Practice-a Survey', Automatica, 1989, 25, (3) pp. 336–337
[2] Vandenboom, T.J.J., Backx, T.C.P.M.: 'Model Predictive Control', (Lecture Notes for the Courses SC4060), Delft, Netherlands, September 2005, pp. 18–21
[3] Kouvaritakis, B., Cannon, M.: 'Model Predictive Control, Classical, Robust and Stochastic', (Springer, 2016)
[4] Juan, J.L., Ling, K., Constantinides G.A., Kerrigan E.C.: 'MPC for Deeply Pipelined FPGA Implementation: Algorithms and Circuitry', IET Control Theory and Applications, 2012, 6, (8) pp. 1029–1041
[5] Ling, K.V., Maciejowski, J.: 'A FPGA Implementation of Model Predictive Control', The American Control Conference, Milwaukee, WI, USA, July 2006, pp. 1–7
[6] Ling, K.V., Wu, B.F., Maciejowski, J.M.: 'Embedded Model Predictive Control (MPC) using a FPGA', The International Federation of Automatic Control, Seoul, Korea, July 2008, pp. 15250–15256
[7] Borrelli, F., Bemporad, A., Morari, M.: 'Predictive Control for Linear and Hybrid Systems', (Cambridge University Press, 2017)
[8] Tondel, P., Johansen, T.A., Bemporad A.: 'An Algorithm for Multi-Parametric Quadratic Programming and Explicit MPC Solutions', Automatica, 2003, 39, pp. 489–497
[9] Rao, C.V., Wright, S.J., Rawlings, J.B.: 'Application of Interior-Point Methods to Model Predictive Control', Journal of Optimization Theory and Applications, 99, (3), December 1998, pp. 723–757
[10] Bemporad, A., Heemels, W.P.M.H., Schutter, B.D.: 'The Explicit Linear Quadratic Regulator for Constrained Systems', Automatica, 2003, 39, pp. 1–16
[11] Fletcher, R.: 'Practical Methods of Optimization', (Wiley, 2nd, 2017)
[12] Xing, C., Tippett, M.J., Xie, L., Bao, J.: 'Fast Distributed MPC based on Active Set Method', Elsevier, 2014, 71, pp. 158–170
[13] Borelli, F., Baotic, M., Pekar, J., Stewart, G.: 'On the Computation of Linear Model Predictive Control Laws', Automatica, 2010, 46, (6), pp. 1035–1041
[14] Wang, Y., Stephen, B.: 'Fast Model Predictive Control Using Online Optimization', IEEE Transactions on Control Systems Technology, 2010, 18, (2), pp. 1–12
[15] Yildirim, E.A., Wright, S.J.:'Warm-Start Strategies in Interior-Point Methods for Linear Programming', SIAM Journal on Optimization, 2002, 12, (3), pp. 782–810
[16] Koehler, S., Danielson, C., Borelli F.: 'A Primal-Dual Active-Set Method for Distributed Model Predictive Control', Optimal Control Applications and Methods, 2017, 38, pp. 339–419

[17] Morari, M.: 'Fast Model Predictive Control', (Lecture Notes for the Courses SC4060), Automatic Control Laboratory, Zurich, Switzerland, September 2013, pp. 1–35
[18] Nicole, Z.M.: 'Real-time Model Predictive Control', Phd Thesis, University of Stutgart, 2011
[19] Zhen, L., Xi, L., Bemporad, A., Lu, S.: 'Fast Linear Parameter Varying Model Predictive Control of Buck DC-DC Converters based on FPGA', IEEE Access, 6, pp. 52434–52446, 2018
[20] Qi, R., Mei, H., Chen, C., Qian, F.: 'A Fast MPC Algorithm for Reducing Computation Burden of MIMO', Elsevier, 23, pp. 2087–2091, 2015
[21] Dunlap, D.D., Collins, E.G., Jr., Caldwell, C.V.: 'Sampling Based Model Predictive Control with Application to Autonomous Vehicle Guidance', Florida Conference on Recent Advances in Robotics, Melbourne, Florida, USA, May 2008, pp. 1–6
[22] Zhang, Y., Nquyen, H., Naidu, D.S., Zou, Y., Cai, C.: 'Time Scale Analysis and Synthesis for Model Predictive Control', International Symposium on Resilient Control Systems, Denver, USA, August 2014, pp. 1–6
[23] Chen, C.: 'Linear System Theory and Design', (Oxford University Press, 3rd, 1999)
[24] DUNDAR, B.: 'Fast Computation Strategies for Model Predictive Control', PhD Thesis (continuing), Dokuz Eylul University, Izmir, Turkey
[25] Cannon, M.: 'C21 Model Predictive Control', (Lecture Notes), Oxford University, Oxford, Term 2016, pp. 1–17
[26] How, J.P.: 'Principles of Optimal Control', (Lecture Notes), MIT Open Courseware, Massachusetts Institue of Technology, Massachusetts, USA, Spring 2008, pp. 1–13