

A Fast Strategy for Model Predictive Control

¹Baris Dundar*, ²Ozge Cihanbegendi

^{1,2}Department of Electrical and Electronics Engineering, Dokuz Eylul University, Tinaztepe, 35390, Buca, Izmir, Turkey

Abstract

This article proposes a method to speed the slow response issues (rise time, settling time and prediction) of the existing MPC algorithms up due to the computation load of online optimization without reducing computation load. In the method, original MPC strategy is warped using an integer time scaling factor in time domain. The aim is to speed this strategy up by exploiting the speeds of fast microprocessors. For this aim, an integer time scaling factor is selected according to the limit of the used microprocessor. Solution of plant, which is controlled, is contracted and sampled by this scaling factor. Standard MPC problem is based on this modification of plant. The first elements of fast computed control law vectors are dilated until original online optimization sampling period is reached again and applied to plant at each sampling period. The modified MPC strategy, which is called Upsampled MPC (UMPC), settles faster to desired output than original MPC strategy and also satisfies faster prediction at each sampling period. Since the method is more compatible with fast microprocessor and the speed of UMPC results from microprocessor, slow operation issue of MPC algorithms is solved for different time scaling factors by this method. Therefore, the control energy of UMPC is lower. There also exists the free times between online optimization sampling periods of original MPC and UMPC strategies. These free times can be used for improvement of the existing MPC algorithms.

Keywords: Model Predictive Control; Plant Model Time Scaling; Fast Settling Response; Free Time.

1. Introduction

In control theory, Model Predictive Control (MPC), which was described by J. Richalet in 1978 for control and stability of dynamical systems (plants), is an online optimal control strategy and it consists of five important parts. These parts are plant model or prediction plant model, objective or cost function, constraints, optimization, and Receding Horizon Principle (RHP) [1-2]. In literature, all the researches focus on these five parts of MPC.

MPC algorithms for Linear Time-Invariant (LTI) dynamical systems require to solve a Quadratic Program (QP) periodically and online using prediction plant model and RHP on constraints at sampling periods. After an MPC problem is converted to a QP, it is solved using improved numerical methods. Thus, current and future inputs (nonlinear control laws) and output responses are predicted. When the existing MPC algorithms contain problems with high-dimensional plant models and constraints, long state and control horizons, and high sampling times, control of plant starts to be difficult [3]. Thus, the MPC algorithms are convenient to be used for slow dynamics because of the computational load, which causes slow response (slow rise time, settling time and prediction), feasibility, stability and timing problems in controlling linear and nonlinear plants with high sampling time. In this case, the classical MPC algorithms are not applicable. The most important reasons for the slow response are as follows:

- Inclusion of physical constraints,
- QP and Nonconvex Nonlinear Programming (NLP) solvers can not converge to a feasible local or global minimum solution and the required time to find a minimum solution can take a long time,
- Non-sparse high-dimensional matrix multiplications and growing computation time with state and control horizons,
- High-dimensional prediction model and physical constrains,

- Control requirements with short sampling periods,
- Time elapsed on the identification, if online model identification is available in control.

The online computational load can be reduced by;

- Using the microprocessors or FPGA ICs, which operate in parallel and faster [4-6],
- Considering the prediction plant model states as the online optimization variable in QP [3],
- Using the improved online and explicit MPC methods (active set, interior point, and multi-parametric solvers) [7-19],
- Reducing dimensions of matrices and vectors and to simplifying multiplication, if it is possible [20],
- Shrinking state and control horizons in the online optimization time, if it is possible [21],
- Implementing the combination of the slow and fast MPC algorithms to plant, which can be divided into slow and fast subsystems [22].

This article focuses on the fast response (i.e. fast rise time, settling time and prediction) of the MPC algorithms. In microprocessors and integrated circuits, control and stability of a plant are controlled at a sampling period T_p , which is selected according to bandwidth and settling time of closed - loop dynamical system or rise time of open - loop dynamical system. T_p is an optimal plant control sampling period for digital control. The existing MPC algorithms solve the QP on discrete - time prediction plant model according to T_p online. In this method, MPC strategy with online optimization sampling period T_p (Original MPC) is warped using an integer time scaling factor $M > 1$ in time domain. The aim is to speed original MPC up by exploiting speeds of fast microprocessors. For this aim, M is selected according to the limit of the microprocessor. The solutions of the continuous-time linear plant are contracted by M . These dynamics will be called as *contracted plant*. The contracted plant is sampled at a new online optimization sampling period $T_m = T_p/M$ to be able to yield the values of the discrete-time plant with T_p , which is a plant model of the original MPC. This discrete-time form of the contracted plant will be called as *Upsampled Plant*. We use the Upsampled Plant as the plant model of the original MPC algorithm. This dynamic will be called as *Upsampled Plant Model (UPM)*. However, all the other components of original MPC algorithm (the cost, reference and constraints functions) are scaled by M and sampled by T_m , as well. The modified MPC operates faster by M times than the plant and original MPC with sampling period T_p due to fast microprocessor and will be called as *Upsampled MPC (UMPC)*. Since UPM has faster sampling period than that of sampling period of the plant model, the term "upsampled" is used. This expression is not related to "upsampling of a signal" in Signal Processing. The first elements of fast computed control law vectors at each T_m in UMPC are dilated till each T_p reaches and applied to plant at each T_p . Thus, the new strategy provides fast response contribution according to the original MPC at each T_m and there are also free times (T_{free}) between the original MPC with T_p and UMPC with T_m at T_p . As another contribution, the free times can be used;

- to compensate modelling errors and disturbances by online system identifications,
- to incorporate with adaptive control and other control strategies,
- to increase control capabilities by different scenarios.

When the free times are not used for the improvement of MPC algorithms, UMPC has energy efficiency according to the original MPC. The method can be applied to the MPC methods [4-22], and extended to nonlinear plant models. In the study, the main idea of the method and its application results on a simulation are discussed.

This article is organized as follows. The UPM is attained in Section 2, the original MPC algorithm with T_p is reformulated for UMPC algorithm with T_m in Section 3, the fast response, free time and lower control energy contributions of UMPC are illustrated on linearized Citation Cessna aircraft dynamics and compared to the original MPC in Section 4, and the conclusions follow in Section 5.

2. Time scaling and upsampled solutions of LTI plant

Consider the deterministic Multi-Input Multi-Output (MIMO) LTI plant is given by the following state equation [23]

$$\begin{aligned}\frac{dx_p(t)}{dt} &= Ax_p(t) + Bu_p(t) \\ y_p(t) &= Cx_p(t) \\ x_p(t_0) &= x_0, \quad \forall t \geq t_0\end{aligned}\tag{1}$$

where state variable $x_p(t) \in \mathbb{R}^n$, initial state variable $x_0 \in \mathbb{R}^n$, input/control variable $u_p(t) \in \mathbb{R}^r$, output variable $y_p(t) \in \mathbb{R}^s$, constant matrices $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times r}$, $C \in \mathbb{R}^{s \times n}$ of the plant (1), and continuous time variable $t \in \mathbb{R}^+$. Assume the plant is controllable.

The solution x_p for the plant (1) is given by

$$\begin{aligned}x_p(t) &= e^{A(t-t_0)}x_0 + \int_{t_0}^t e^{A(t-\tau)}Bu_p(\tau)d\tau \\ &\triangleq x_{px}(t) + x_{pu}(t)\end{aligned}\tag{2}$$

$x_p(t)$ is a linear combination of $x_{px}(t)$ and $x_{pu}(t)$ resulting from superposition theorem. $x_{px}(t)$ is zero-input response with non-zero initial state x_0 and $x_{pu}(t)$ is zero-state with non-zero input.

Consider a microprocessor, which can operate faster than online optimization sampling period of original MPC, and under this condition, we can select a large enough integer time scaling factor $M > 1$.

Then we can define the following dynamics for time-scaled solutions of plant (1)

$$\begin{aligned}\frac{dx_m(t)}{dt} &= M(Ax_m(t) + Bu_m(t)) \\ y_m(t) &= Cx_m(t) \\ x_m(t_0) &= x_0, \quad \forall t \geq t_0\end{aligned}\tag{3}$$

where state variable $x_m(t) \in \mathbb{R}^n$, the measured or estimated initial state variable $x_0 \in \mathbb{R}^n$ from the plant (1), input/control variable $u_m(t) \in \mathbb{R}^r$, output variable $y_m(t) \in \mathbb{R}^s$ of the dynamics (3). The time-scaled state space dynamics are called *contracted plant* in time domain. Multiplying $e^{-MA t}$ on both side of state equation of the contracted plant (2), the following equation is yielded

$$e^{-MA t} \frac{dx_m(t)}{dt} - Me^{-MA t} Ax_m(t) = Me^{-MA t} Bu_m(t)\tag{4}$$

This equation is the same to the following equation

$$\frac{d}{dt}(e^{-MA t} x_m(t)) = Me^{-MA t} Bu_m(t)\tag{5}$$

Integrating both side of the equation from t_0 to t

$$e^{-MA t} x_m(t)|_{\tau=t_0}^t = M \int_{t_0}^t e^{-MA \tau} Bu_m(\tau) d\tau\tag{6}$$

we reach

$$e^{-MA t} x_m(t) - e^{-MA t_0} x_0 = M \int_{t_0}^t e^{-MA \tau} Bu_m(\tau) d\tau\tag{7}$$

Thus we have the solution $x_m(t)$ of contracted plant

$$x_m(t) = x_0 e^{MA(t-t_0)} + M \int_{t_0}^t e^{MA(t-\tau)} Bu_m(\tau) d\tau\tag{8}$$

Using change of $t = Mt'$, $t_0 = Mt'_0$ and $\tau = M\tau'$ for the solution (2)

$$\underbrace{x_p(t)}_{t=Mt'} = \underbrace{x_0 e^{A(t-t_0)}}_{t=Mt', t_0=Mt'_0} + \underbrace{e^{At}}_{t=Mt'} \int_{t_0}^t \underbrace{e^{-A\tau} B u_p(\tau) d\tau}_{\tau=M\tau'} \quad (9)$$

we can write the time scaling relation between the solutions of the plant $x_p(t)$ and the contracted plant $x_m(t)$

$$x_m(t) = x_p(Mt) \quad (10)$$

under the condition

$$u_m(t) = u_p(Mt) \quad (11)$$

$x_m(t)$ has faster response by M times than $x_p(t)$ in time domain. Consider the discrete-time form of the plant (1) is sampled at T_p of the original MPC algorithm.

$$\begin{aligned} x_p(k+1) &= A_{pd}x_p(k) + B_{pd}u_p(k) \\ y_p(k) &= Cx_p(k) \\ x_p(0) &= x_0, \quad \forall k \geq 0 \end{aligned} \quad (12)$$

where $k := kT_p$ for time instant $k = 0, 1, 2, \dots$, $u_p(kT_p) := u_p(k)$ and

$$\begin{aligned} A_{pd} &= e^{AT_p} \\ B_{pd} &= \left(\int_0^{T_p} e^{A\tau} d\tau \right) B \end{aligned} \quad (13)$$

The solution of the discrete-time plant (12) is given by

$$\begin{aligned} x_p(k) &= A_{pd}^k x_0 + \sum_{m=0}^{k-1} A_{pd}^{k-1-m} B_{pd} u_p(m) \\ &\triangleq x_{px}(k) + x_{pu}(k) \end{aligned} \quad (14)$$

The discrete-time plant (12) with T_p is the plant model of the original MPC algorithm with T_p .

We claim that the original MPC algorithm is contracted by M and has a faster online optimization period $T_m = T_p/M$. This algorithm will be called as UMPC. For this aim, the contracted plant (3) is sampled by $T_m = T_p/M$. This discrete-time form is yielded as

$$\begin{aligned} x_m(k+1) &= A_{md}x_m(k) + B_{md}u_m(k) \\ y_m(k) &= Cx_m(k) \\ x_m(0) &= x_0, \quad \forall k \geq 0 \end{aligned} \quad (15)$$

where $k := kT_m$ for time instant $k = 0, 1, \dots$, $u_m(kT_m) := u_m(k)$ and

$$\begin{aligned} A_{md} &= e^{MAT_m} = A_{pd} \\ B_{md} &= M \left(\int_0^{T_m} e^{MA\tau} d\tau \right) B = B_{pd} \end{aligned} \quad (16)$$

Note that $B_{md} = B_{pd}$ can be yielded using change of variables. The solution of the discrete-time plant model (15) is given by

$$\begin{aligned} x_m(k) &= A_{md}^k x_0 + \sum_{m=0}^{k-1} A_{md}^{k-1-m} B_{md} u_m(m) \\ &\triangleq x_{mx}(k) + x_{mu}(k) \end{aligned} \quad (17)$$

The equation (15) is called *Upsampled Plant Model (UPM)* of UMPC.

The dynamic pairs (12) and (15) have the same behavior at different times.

We can write the following time-shifted relationship between $x_p(k)$ and $x_m(k)$.

$$x_p(kT_p) = x_m(k(T_m + T_{free})) \quad \forall k \quad (18)$$

where

$$\begin{aligned} T_{free} &= T_p - T_m \\ &= T_m(M - 1) \end{aligned} \quad (19)$$

We can state from the equation (18) that if the values of $x_m(kT_m)$ are dilated by T_{free} by saving its values at each time instant k , it is equated to $x_p(kT_p)$. In this case, the UPM (15) is transformed to the original plant model (12).

2.1 Case study

2.1.1 Case 1

Consider the feedback linearized simple pendulum

$$\begin{aligned} \dot{x}_{p1}(t) &= x_{p2}(t) \\ \dot{x}_{p2}(t) &= u_p(t) \\ x_p(0) &= x_0 = [x_{01} \ x_{02}]^T, \quad \forall t \geq 0 \end{aligned} \quad (20)$$

where $x_1(t)$ is rad/sn.

For $u_p(t) = \sin(t)$, the solution is

$$\begin{aligned} x_{p1}(t) &= x_{01} + (x_{02} + 1)t - \sin(t) \\ x_{p2}(t) &= x_{02} - \cos(t) + 1 \end{aligned} \quad (21)$$

The contracted plant is

$$\begin{aligned} \dot{x}_{m1}(t) &= Mx_{m2}(t) \\ \dot{x}_{m2}(t) &= Mu_m(t) \\ x_m(0) &= x_0 = [x_{01} \ x_{02}]^T, \quad \forall t \geq 0 \end{aligned} \quad (22)$$

For $u_p(t) = \sin(t)$, the solution is

$$\begin{aligned} x_{m1}(t) &= x_{01} + (x_{02} + 1)Mt - \sin(Mt) \\ x_{m2}(t) &= x_{02} - \cos(Mt) + 1 \end{aligned} \quad (23)$$

For $u_p(t) = t + t\sin(t)$, the solutions are

$$\begin{aligned} x_{p1}(t) &= x_{01} + x_{02}t + 1/6t^3 - t\sin(t) - 2\cos(t) + 2 \\ x_{p2}(t) &= x_{02} + 1/2t^2 + \sin(t) - t\cos(t) \end{aligned} \quad (24)$$

$$\begin{aligned} x_{m1}(t) &= x_{01} + x_{02}Mt + 1/6(Mt)^3 - Mtsin(Mt) \\ &\quad - 2\cos(Mt) + 2 \\ x_{m2}(t) &= x_{02} + 1/2(Mt)^2 + sin(Mt) - Mtcos(Mt) \end{aligned} \quad (25)$$

The solution pairs (21-23) and (24-25) satisfy the equation (10) under the condition (11) [24].

2.1.2 Case 2

Consider the linearized continuous-time plant (Citation Aircraft)

$$\dot{x}_p(t) = \begin{bmatrix} -1.2822 & 0 & 0.98 & 0 \\ 0 & 0 & 1 & 0 \\ -5.4293 & 0 & -1.8366 & 0 \\ -128.2 & 128.2 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_{p1}(t) \\ x_{p2}(t) \\ x_{p3}(t) \\ x_{p4}(t) \end{bmatrix} + \begin{bmatrix} -0.3 \\ 0 \\ -17 \\ 0 \end{bmatrix} u_p(t)$$

$$y_p(t) = \begin{bmatrix} y_{p1}(t) \\ y_{p2}(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{p1}(t) \\ x_{p2}(t) \\ x_{p3}(t) \\ x_{p4}(t) \end{bmatrix}$$

$$x_p(0) = x_0, \quad \forall t \geq 0$$
(26)

where $u_p(t)$ is the elevator angle, $x_{p1}(t)$ is the angle of attack, $x_{p2}(t)$ is the pitch angle, $x_{p3}(t)$ is the pitch rate, $x_{p4}(t)$ is the altitude, the outputs $y_{p1}(t)$ and $y_{p2}(t)$ are the pitch angle and the altitude, respectively.

The constraints are the elevator angle $-0.262 \leq u(t) \leq 0.262 \text{ rad.}$, the elevator rate $-0.0698 \leq \Delta u_p(t) \leq 0.0698 \text{ rad.}$, the pitch angle $-0.349 \leq y_{p1}(t) \leq 0.349 \text{ rad.}$ The plant is linearized at altitude of 5000 m. and a speed of 128.2 m. / sec. The open-loop impulse response vector $x_p(t)$ is unstable due to open-loop poles are $0, 0, -1.5594 \pm 2.29j$ [Zeilinger et al, 2016].

The plant model of the original MPC for the plant (26) at the online optimization sampling period $T_p = 0.2 \text{ sec.}$ is

$$x_p(k+1) = \begin{bmatrix} 0.6958 & 0 & 0.1385 & 0 \\ -0.08697 & 1 & 0.1619 & 0 \\ -0.7674 & 0 & 0.6174 & 0 \\ -22.68 & 25.64 & 0.2272 & 1 \end{bmatrix} \begin{bmatrix} x_{p1}(k) \\ x_{p2}(k) \\ x_{p3}(k) \\ x_{p4}(k) \end{bmatrix} + \begin{bmatrix} -0.3181 \\ -0.2952 \\ -2.726 \\ 0.4953 \end{bmatrix} u_p(k)$$

$$y_p(k) = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{p1}(k) \\ x_{p2}(k) \\ x_{p3}(k) \\ x_{p4}(k) \end{bmatrix}$$

$$x_p(0) = x_0, \quad \forall k \geq 0$$
(27)

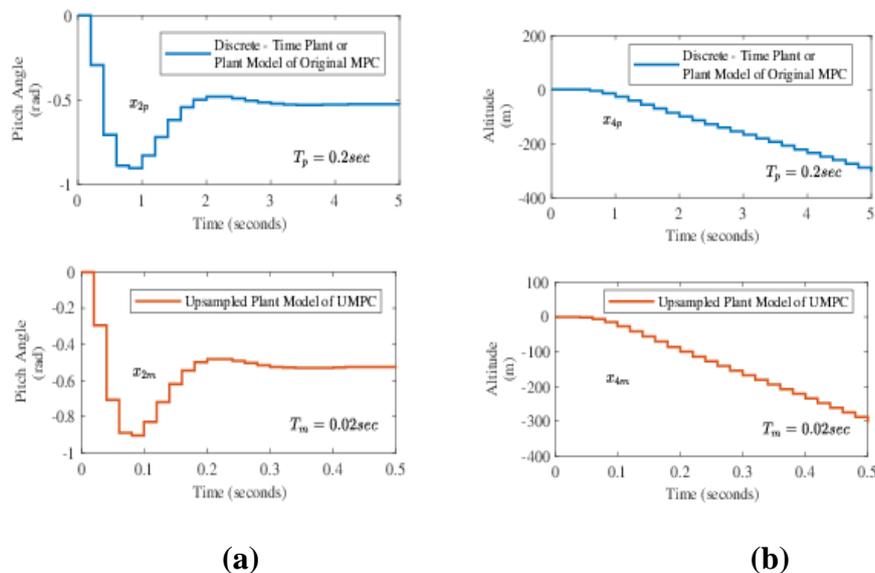


Fig.1 Discrete - Time Plant or Plant Model of Original MPC and UPM of UMPC Impulse Responses (a), The Pitch Angle Impulse Responses (b), The Altitude Impulse Responses

The contracted plant for $M = 10$ is

$$\dot{x}_m(t) = 10 \begin{bmatrix} -1.2822 & 0 & 0.98 & 0 \\ 0 & 0 & 1 & 0 \\ -5.4293 & 0 & -1.8366 & 0 \\ -128.2 & 128.2 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_{m1}(t) \\ x_{m2}(t) \\ x_{m3}(t) \\ x_{m4}(t) \end{bmatrix} + 10 \begin{bmatrix} -0.3 \\ 0 \\ -17 \\ 0 \end{bmatrix} u_m(t) \quad (28)$$

$$y_m(t) = \begin{bmatrix} y_{m1}(t) \\ y_{m2}(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{m1}(t) \\ x_{m2}(t) \\ x_{m3}(t) \\ x_{m4}(t) \end{bmatrix}$$

$$x_m(0) = x_0, \quad \forall t \geq 0$$

where $u_m(t) = u_p(10t)$. The UPM of UMPC at $T_m = 0.02 \text{ sec.}$ is

$$x_m(k+1) = \begin{bmatrix} 0.6958 & 0 & 0.1385 & 0 \\ -0.08697 & 1 & 0.1619 & 0 \\ -0.7674 & 0 & 0.6174 & 0 \\ -22.68 & 25.64 & 0.2272 & 1 \end{bmatrix} \begin{bmatrix} x_{m1}(k) \\ x_{m2}(k) \\ x_{m3}(k) \\ x_{m4}(k) \end{bmatrix} + \begin{bmatrix} -0.3181 \\ -0.2952 \\ -2.726 \\ 0.4953 \end{bmatrix} u_m(k) \quad (29)$$

$$y_m(k) = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{m1}(k) \\ x_{m2}(k) \\ x_{m3}(k) \\ x_{m4}(k) \end{bmatrix}$$

$$x_m(0) = x_0, \quad \forall k \geq 0$$

The equations (27) and (29) satisfy the equation (18). The impulse responses are illustrated in Fig. 1. As the plant model reaches a value in 5 sec., the UPM reaches the same value in 0.5 sec.

The free time between the plant model (27) of MPC and the UPM (29) of UMPC are $T_{free} = 0.18 \text{ sec.}$ at each $T_p = 0.2 \text{ sec.}$ If the UPM (23) of UMPC is dilated by $T_{free} = 0.18 \text{ sec.}$ at each $T_m = 0.02 \text{ sec.}$, it is equated to plant model (27) of the original MPC with $T_p = 0.2 \text{ sec.}$

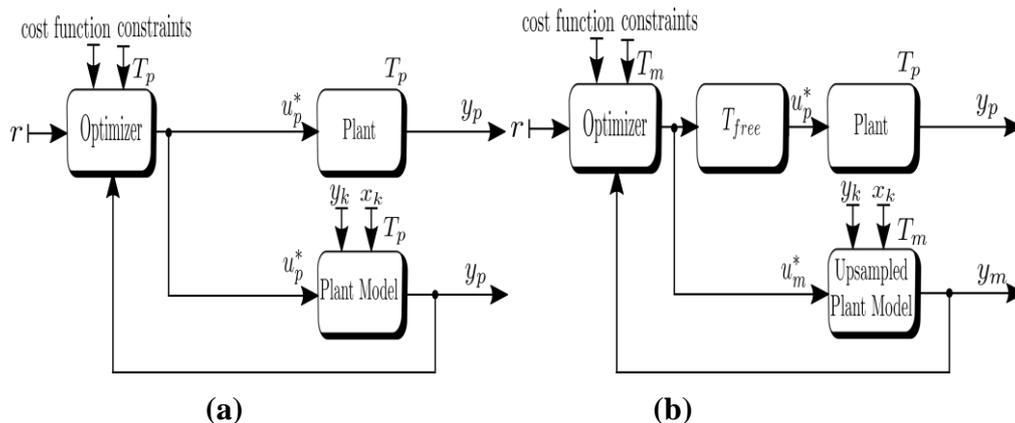


Fig.2 Equivalent MPC Controller Block Structures (a), Original MPC Block Structure (b), UMPC Block Structure

3. Nominal UMPC

We establish a nominal or standard predictive control algorithm based on the UPM and call it as the Original MPC. Assume the plant (1) or (12) must be controlled by it, which has the plant model (12) at the online optimization sampling period T_p and does not have an error between the plant and the plant model. We want to speed response (rise time, settling time and prediction) of the original MPC up. For this aim, the original

MPC algorithm is reformulated based on the UPM (15) instead of the plant model (12) and the online optimization sampling period becomes T_m . This MPC approach is called as Upsampled MPC. The MPC and UMPC block diagrams are illustrated in Fig. 2.

The prediction UPM for the UPM (15) with T_m is

$$\begin{aligned} x_m(k+i+1) &= A_{md}x_m(k+i) + B_{md}\Delta u_m(k+i) \\ y_m(k+i) &= Cx_m(k+i) \\ x_m(k) &= x_k, \quad \forall k, i \geq 0 \end{aligned} \quad (30)$$

where x_k is the measured or estimated initial state value from the plant (1) and then sampled by T_m at k and $\Delta u_m(k) = u_m(k) - u_m(k-1)$ denotes the control increments or the input rate. These forms of control laws contain automatic integral action in feedback loop and are used for minimum over - shoot and settling time in the online optimization process. Note that integral action rejects constant additive disturbances in the steady state. The computations are realized 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 prediction equation at k is found as

$$\mathbf{x}_m(k) = \Theta x_k + \Phi \Delta \mathbf{u}_m(k) \quad (31)$$

where

$$\mathbf{x}_m(k) = \begin{bmatrix} x_m(k+1) \\ x_m(k+2) \\ \vdots \\ x_m(k+N) \end{bmatrix}^T \in \mathbb{R}^{Nn} \quad (32)$$

$$\Delta \mathbf{u}_m(k) = \begin{bmatrix} \Delta u_m(k) \\ \Delta u_m(k+1) \\ \vdots \\ \Delta u_m(k+N-1) \end{bmatrix}^T \in \mathbb{R}^{Nr}$$

$$\Theta = [A_{md} \quad A_{md}^2 \quad \dots \quad A_{md}^N]^T \in \mathbb{R}^{Nn \times n} \quad (33)$$

$$\begin{aligned} \Phi &= [\Phi_1 \quad \Phi_2 \quad \dots \quad \Phi_N]^T \\ &= \begin{bmatrix} B_{md} & 0 & \dots & 0 \\ A_{md}B_{md} & B_{md} & & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A_{md}^{N-1}B_{md} & A_{md}^{N-2}B_{md} & & B_{md} \end{bmatrix} \in \mathbb{R}^{Nn \times Nr} \end{aligned} \quad (34)$$

The prediction UPM (30) is subject to the linear state and input rate prediction constraints with T_m

$$\begin{aligned} \Delta u_{min} &\leq \Delta u_m(k+i) \leq \Delta u_{max} \\ x_{min} &\leq x_m(k+i) \leq x_{max} \end{aligned} \quad (35)$$

We can now express the following UMPC problem with T_m instead of the original MPC problem with T_p for controlling the plant (1) [25].

$$\begin{aligned}
J_k^*(x_k) &= \min_{\Delta \mathbf{u}_m(k)} J_k(x_k, \Delta \mathbf{u}_m(k)) \\
&= \sum_{i=0}^{N-1} [x_m^T(k+i)q_w x_m(k+i) + \Delta u_m^T(k+i) \\
&\quad \times r_w \Delta u_m(k+i) + x_m^T(k+N)p_N x_m(k+N)] \\
&\text{subject to} \\
\Delta u_{min} &\leq \Delta u_m(k+i) \leq \Delta u_{max} & i = 0, 1, \dots, N-1 \quad \forall k \\
x_{min} &\leq x_m(k+i) \leq x_{max} & i = 1, \dots, N-1 \quad \forall k \\
\Delta u_{min} &\leq K(A_{md} + B_{md}K)^i x_m(k+N) \leq \Delta u_{max} & i = 0, 1, \dots, \varepsilon \quad \forall k \\
x_{min} &\leq (A_{md} + B_{md}K)^i x_m(k+N) \leq x_{max} & i = 0, 1, \dots, \varepsilon \quad \forall k \\
x_m(k) &= x_k \quad \text{at } k
\end{aligned} \tag{36}$$

where $J_k(x_k, \Delta \mathbf{u}_m(k))$ 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 the positive definite. This problem for the plant model (12) is the original MPC problem. The state feedback gain K is computed on the infinite horizon and unconstrained form (Linear Quadratic Optimal Control Problem) of the UMPC algorithm (36). The following conditions must be satisfied to be able to ensure closed-loop stability:

- If the pair (A_{md}, B_{md}) is stabilizable (controllable), the pair (A_{md}, q_w) is observable and r_w is positive-definite, the state feedback gain $K \in \mathbb{R}^{r \times n}$ is found offline by the following discrete-time Algebraic Ricatti Equation

$$K = -(B_{md}^T P_{\infty} B_{md} + R_w)^{-1} B_{md} P_{\infty} A_{md} \tag{37}$$

where the unique and positive matrix P_{∞} is found by the following matrix equation

$$P_{\infty} = A_{md}^T P_{\infty} A_{md} + Q'_w - A_{md}^T P_{\infty} B_{md} \times (B_{md}^T P_{\infty} B_{md} + R_w)^{-1} B_{md}^T P_{\infty} A_{md} \tag{38}$$

The weighted matrices are given by

$$Q'_w = \begin{bmatrix} q_w & 0 & \dots & 0 \\ 0 & \ddots & \dots & \vdots \\ \vdots & \dots & q_w & 0 \\ 0 & \dots & 0 & q_w \end{bmatrix} \in \mathbb{R}^{Nn \times Nn} \tag{39}$$

$$R_w = \begin{bmatrix} r_w & 0 & \dots & 0 \\ 0 & \ddots & \dots & \vdots \\ \vdots & \dots & r_w & 0 \\ 0 & \dots & 0 & r_w \end{bmatrix} \in \mathbb{R}^{Nr \times Nr} \tag{40}$$

- The positive definite matrix p_N of the terminal cost function $x(k+N)^T p_N x(k+N)$ must be found offline by the following unique Lyapunov Matrix Equation

$$\begin{aligned}
p_N - (A_{md} + B_{md}K)^T p_N (A_{md} + B_{md}K) \\
= q_w + K^T r_w K
\end{aligned} \tag{41}$$

- The minimum horizon ε of the terminal constraints for recursively feasibility is found offline by the following linear programs

$$\begin{aligned} \Delta u_{max,j} &= \max_x K_j (A_{md} + B_{md}K)^{\varepsilon+1} x \\ \text{s. t.} & \\ \Delta u_{min} &\leq K(A_{md} + B_{md}K)^i x \leq \Delta u_{max} \\ i &= 0,1,\dots,\varepsilon \end{aligned} \tag{42}$$

and

$$\begin{aligned} \Delta u_{min,j} &= \min_x K_j (A_{md} + B_{md}K)^{\varepsilon+1} x \\ \text{s. t.} & \\ \Delta u_{min} &\leq K(A_{md} + B_{md}K)^i x \leq \Delta u_{max} \\ i &= 0,1,\dots,\varepsilon \end{aligned} \tag{43}$$

where any state $x \in \mathbb{R}^n$ and $j = 0,1,\dots,r$ of the input $u(k) \in \mathbb{R}^r$.

The constraints at k are written in terms of $\Delta \mathbf{u}_m$

$$\begin{bmatrix} \Delta u_{min} \\ \vdots \\ \Delta u_{min} \\ x_{min} \\ \vdots \\ x_{min} \\ \Delta u_{min} \\ \vdots \\ \Delta u_{min} \\ x_{min} \\ \vdots \\ x_{min} \end{bmatrix} \leq \begin{bmatrix} \Delta u_m(k) \\ \vdots \\ \Delta u_m(k+N-1) \\ x_m(k+1) \\ \vdots \\ x_m(k+N-1) \\ Kx_m(k+N) \\ \vdots \\ K(A+BK)^\varepsilon x_m(k+N) \\ x_m(k+N) \\ \vdots \\ (A+BK)^\varepsilon x_m(k+N) \end{bmatrix} \leq \begin{bmatrix} \Delta u_{max} \\ \vdots \\ \Delta u_{max} \\ x_{max} \\ \vdots \\ x_{max} \\ \Delta u_{max} \\ \vdots \\ \Delta u_{max} \\ x_{max} \\ \vdots \\ x_{max} \end{bmatrix} \tag{44}$$

The constraint (44) is converted in the following compact inequality form

$$F \Delta \mathbf{u}_m(k) \leq G \tag{45}$$

All the matrices in the inequality constraint (45) are

$$F = \begin{bmatrix} F_2 \\ -F_2 \end{bmatrix} \in \mathbb{R}^{2[(N+\varepsilon+1)r+(N+\varepsilon)n] \times Nr}, \tag{46}$$

$$G = \begin{bmatrix} F_{max} - F_1 x_k \\ -F_{min} + F_1 x_k \end{bmatrix} \in \mathbb{R}^{2[(N+\varepsilon+1)r+(N+\varepsilon)n]}, \tag{47}$$

$$F_{min} = \begin{bmatrix} \Delta u_{min} \\ \vdots \\ \Delta u_{min} \\ x_{min} \\ \vdots \\ x_{min} \\ \Delta u_{min} \\ \vdots \\ \Delta u_{min} \\ x_{min} \\ \vdots \\ x_{min} \end{bmatrix}, F_{max} = \begin{bmatrix} \Delta u_{max} \\ \vdots \\ \Delta u_{max} \\ x_{max} \\ \vdots \\ x_{max} \\ \Delta u_{max} \\ \vdots \\ \Delta u_{max} \\ x_{max} \\ \vdots \\ x_{max} \end{bmatrix}, \tag{48}$$

$$F_1 = \begin{bmatrix} F_{11} \\ F_{12} \\ F_{13} \\ F_{14} \end{bmatrix}, F_2 = \begin{bmatrix} F_{21} \\ F_{22} \\ F_{23} \\ F_{24} \end{bmatrix}, \quad (49)$$

$$F_{11} = [\mathbf{0}] \in \mathbb{R}^{Nr \times n}, \quad (50)$$

$$F_{12} = [A_{md} \quad A_{md}^2 \quad \dots \quad A_{md}^{N-1}]^T \in \mathbb{R}^{(N-1)n \times n}, \quad (51)$$

$$F_{13} = \begin{bmatrix} KA_{md}^N \\ \vdots \\ K(A_{md} + B_{md}K)^\varepsilon A_{md}^N \end{bmatrix} \in \mathbb{R}^{(\varepsilon+1)r \times n}, \quad (52)$$

$$F_{14} = \begin{bmatrix} A_{md}^N \\ \vdots \\ (A_{md} + B_{md}K)^\varepsilon A_{md}^N \end{bmatrix} \in \mathbb{R}^{(\varepsilon+1)n \times n}, \quad (53)$$

$$F_{21} = [\mathbf{I}] \in \mathbb{R}^{Nr \times Nr}, \quad (54)$$

$$F_{22} = [\Phi_1 \quad \Phi_2 \quad \dots \quad \Phi_{N-1}]^T \in \mathbb{R}^{(N-1)n \times Nr}, \quad (55)$$

$$F_{23} = \begin{bmatrix} a & & b \\ \vdots & \ddots & \\ c & & d \end{bmatrix} \in \mathbb{R}^{(\varepsilon+1)r \times Nr} \quad (56)$$

$$a = KA_{md}^{N-1}B, \quad (57)$$

$$b = KB_{md}, \quad (58)$$

$$c = K(A_{md} + B_{md}K)^\varepsilon A_{md}^{N-1}B_{md}, \quad (59)$$

$$d = K(A_{md} + B_{md}K)^\varepsilon B_{md}, \quad (60)$$

$$F_{24} = \begin{bmatrix} f & & g \\ \vdots & \ddots & \\ h & & i \end{bmatrix} \in \mathbb{R}^{(\varepsilon+1)n \times Nr} \quad (61)$$

$$f = A_{md}^{N-1}B, \quad (62)$$

$$g = B_{md}, \quad (63)$$

$$h = (A_{md} + B_{md}K)^\varepsilon A_{md}^{N-1} B_{md}, \quad (64)$$

$$i = (A_{md} + B_{md}K)^\varepsilon B_{md}. \quad (65)$$

The matrices F and G are computed offline.

The UMPC problem at k is converted to the following QP in terms of $\Delta \mathbf{u}_m$

$$\begin{aligned} J_k^*(x_k) &= \min_{\Delta \mathbf{u}_m(k)} J_k(x_k, \Delta \mathbf{u}_m(k)) \\ &= \Delta \mathbf{u}_m^T(k) V \Delta \mathbf{u}_m(k) + 2x_k^T Y^T \Delta \mathbf{u}_m + x_k^T Z x_k \\ &\text{subject to} \\ &F \Delta \mathbf{u}_m(k) \leq G \end{aligned} \quad (66)$$

where $k := kT_m$, $k = 0, 1, \dots$

The matrices in the QP (66) are given by

$$\begin{aligned} V &= \Phi^T Q_w \Phi + R_w \\ Y &= \Phi^T Q_w \Theta \\ Z &= \Theta^T Q_w \Theta + q_w \end{aligned} \quad (67)$$

$$Q_w = \begin{bmatrix} q_w & 0 & \dots & 0 \\ 0 & \ddots & \dots & \vdots \\ \vdots & \dots & q_w & 0 \\ 0 & \dots & 0 & p_N \end{bmatrix} \in \mathbb{R}^{Nn \times Nn} \quad (68)$$

$$R_w = \begin{bmatrix} r_w & 0 & \dots & 0 \\ 0 & \ddots & \dots & \vdots \\ \vdots & \dots & r_w & 0 \\ 0 & \dots & 0 & r_w \end{bmatrix} \in \mathbb{R}^{Nr \times Nr} \quad (69)$$

The Lagrangian cost function for the QP (66) is

$$\begin{aligned} \mathbf{J}(\Delta \mathbf{u}_m(k), \lambda) &= \Delta \mathbf{u}_m^T(k) V \Delta \mathbf{u}_m(k) + 2x_k^T Y^T \Delta \mathbf{u}_m(k) \\ &+ x_k^T Z x_k + \lambda^T (F \Delta \mathbf{u}_m(k) + G - h) \end{aligned} \quad (70)$$

where $\lambda \in \mathbb{R}^{Nr}$ and $h \in \mathbb{R}^{2[(N+\varepsilon+1)r+(N+\varepsilon)n]}$ are the Lagrange multiplier and the slack variable, respectively. The minimum of the Lagrangian cost function (70) is found by the following Karush-Kuhn-Tucker (KKK) conditions.

$$\frac{\partial \mathbf{J}}{\partial \Delta \mathbf{u}_m} = 2V \Delta \mathbf{u}_m(k) + 2Y x_k + F^T \lambda = 0, \quad (71)$$

$$\frac{\partial \mathbf{J}}{\partial \lambda} = F \Delta \mathbf{u}_m(k) + G - h = 0, \quad (72)$$

$$\lambda \geq 0, \quad (73)$$

$$h \geq 0, \quad (74)$$

$$h^T \lambda = 0. \quad (75)$$

The conditions (71-72) and (73-75) are called as feasibility and complementarity conditions, respectively. Since the QP (70) has the convex cost function and the linear constraints, the computations in KKK conditions (71-75) are concluded the global minimum with the feasible initial state. Thus, we can generally write the plant model optimal control law vector $\Delta \mathbf{u}_m^*(k)$ as

$$\begin{aligned} \Delta \mathbf{u}_m^*(k) &= \underset{\Delta \mathbf{u}_m(k)}{\operatorname{arg\,min}} J_k(x_k, \Delta \mathbf{u}_m(k)) \\ &\stackrel{\Delta}{=} F_N * x_k \end{aligned} \quad (76)$$

over the horizon N at time instant k . $\Delta \mathbf{u}_m^*$ is computed online using the improved QP solvers and the nonlinear control laws are computed over the horizon N for any k . If the UMPC problem is written for plant model (12), it is the original MPC problem and is converted to the following QP with T_p at k

$$\begin{aligned} J_k^*(x_k) &= \min_{\Delta \mathbf{u}_p(k)} J_k(x_k, \Delta \mathbf{u}_p(k)) \\ &= \Delta \mathbf{u}_p^T(k) V \Delta \mathbf{u}_p(k) + 2x_k^T(k) Y^T \Delta \mathbf{u}_p \\ &\quad + x_k^T Z x_k \\ &\text{subject to} \\ &F \Delta \mathbf{u}_p(k) \leq G \end{aligned} \quad (77)$$

where $k: = kT_p, k = 0, 1, \dots$

The plant optimal control law vector $\Delta \mathbf{u}_p^*(k)$ is

$$\begin{aligned} \Delta \mathbf{u}_p^*(k) &= \underset{\Delta \mathbf{u}_p(k)}{\operatorname{arg\,min}} J_k(x_k, \Delta \mathbf{u}_p(k)) \\ &\stackrel{\Delta}{=} F_N * x_k \end{aligned} \quad (78)$$

over the horizon N at k . The QPs (66) and (77) are the same but instead of $\Delta \mathbf{u}_p^*(k)$ at k , which must be implemented to the plant (1) or (12), $\Delta \mathbf{u}_m^*(k)$ is implemented and it is more compatible to microprocessors, which operate fast and parallel.

Thus, we can write the time-shifted relationship between the control laws of MPC and UMPC algorithms for any k as

$$\Delta \mathbf{u}_p^*(kT_p) = \Delta \mathbf{u}_m^*(k(T_m + T_{free})) \quad \forall k \quad (79)$$

and using $\Delta u_m(k) = u_m(k) - u_m(k - 1)$,

$$\mathbf{u}_p^*(kT_p) = \mathbf{u}_m^*(k(T_m + T_{free})) \quad \forall k \quad (80)$$

is computed. M should be selected for the desired T_{free} . The values of $\Delta \mathbf{u}_m^*(kT_m)$ are dilated by the free time T_{free} by saving its values in registers of microprocessor and then implemented to the plant at kT_p until next value $(k + 1)T_p$ at each sampling instant k . Thus, $\Delta \mathbf{u}_m^*(kT_m)$ can be used instead of $\Delta \mathbf{u}_p^*(kT_m)$. Finally, we achieve fast response and free times using UMPC algorithm instead of the original MPC algorithm. The predicted responses in the original MPC and UMPC at $k = 1$ are illustrated for controlling plant (1) or (12) in Fig. 3. This process is repeated till the reference function on receding horizon N .

Our analysis leads to the algorithm in the Algorithm 1.

UMPC Algorithm

1. Measure the state x_k at kT_m
2. Compute $\Delta \mathbf{u}_m^*(kT_m)$ and $\mathbf{u}_m^*(kT_m)$ at kT_m
3. If $\Delta \mathbf{u}_m^*(kT_m) = 0$ Then 'Problem Infeasible' Stop

4. Wait or Improve MPC Algorithm in $k(M - 1)T_m = kT_{free}$
5. Implement the first element of $\mathbf{u}_m^*(kT_m)$ to the plant at $kMT_m (= kT_p)$
6. Wait for the new sampling time $(k + 1)MT_m (= (k + 1)T_p)$ Go to 1

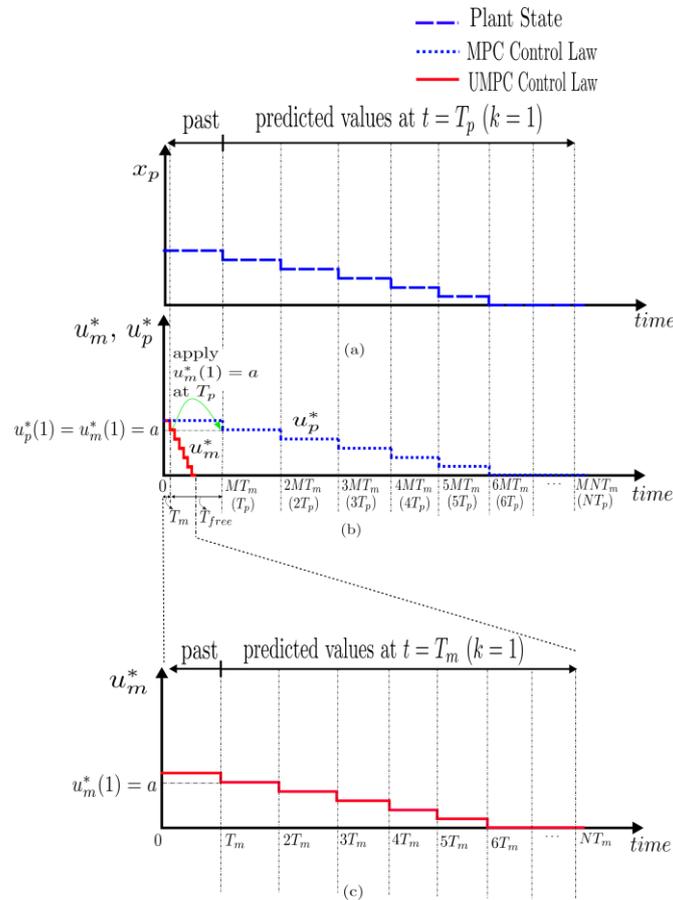


Fig. 3 The plant control using Original MPC and UMPC (a), The Predicted Plant State Response at $T_p (k = 1)$ (b), The Predicted Input Response at $T_p (k = 1)$ using Original MPC (c), The Fast Predicted Input Response at $T_m (k = 1)$ using UMPC

3.1 Selecting time scaling factor M

In this article, we state that the plant must be controlled at T_p , which is selected according to the Nyquist-Shannon Sampling Theorem [26]. In the UMPC algorithm, we claim the maximum M (i.e. the minimum T_m instead T_p).

In real time, M depends on the physical constrains (such as speeds of microprocessors, integrated circuits, ADC and DAC , etc.).

If a microprocessor or integrated circuit operates at the sampling period $T_{\mu P}$, maximum M can be selected as the following proportion

$$M = T_p/T_{\mu P} \quad \text{for integer} \quad M > 1 \quad (81)$$

The sufficient condition for the above proportion is that sampled values at T_p must be taken at T_m from the plant, as well. Thus, the constraint for maximum M can be selected as $T_{\mu P}$. Hence, the maximum online optimization sampling period for the UMPC algorithm can be selected as

$$T_m = T_{\mu P} = T_p/M \quad \text{for integer} \quad M > 1 \quad (82)$$

Besides, the sampling period of the plant model (12) can be selected as

$$T'_p = T_p/L \quad \text{for integer } L \quad (83)$$

This equation provides sufficient condition to use the previously computed sequence of the optimal control law u_p^* at current time. In this article, the sampling period of the plant model (1) is assumed as $T'_p = T_p$.

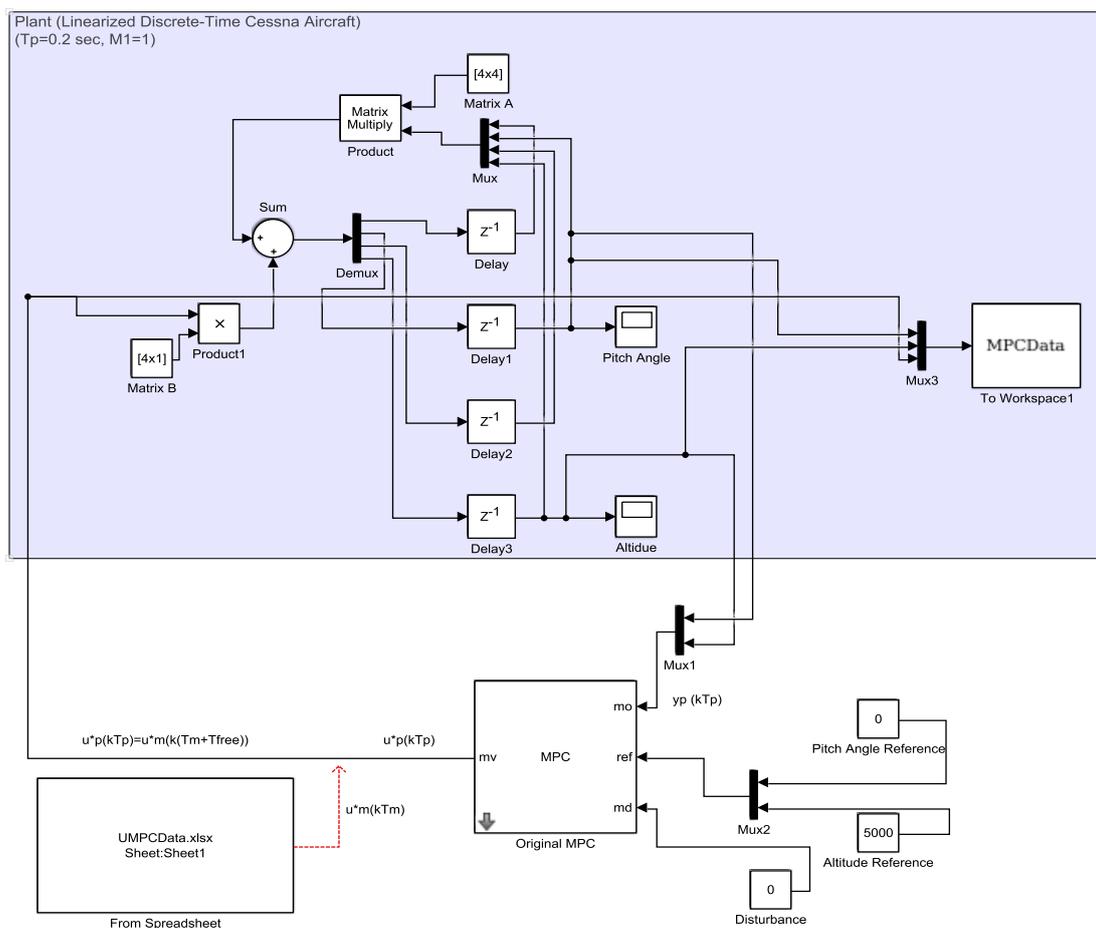
4. Simulation results and discussion

The plant (26) or its discrete-time form (27) must be controlled at $T_p = 0.2 \text{ sec.}$ using the original MPC ($M = 1$) with the plant model (27). The MPC and UMPC systems in Fig.4 are established to illustrate the application of proposed method on the plant (27). The Fig.4(a) and (b) represent the original MPC and UMPC systems. For the application of method using Model Predictive Control Toolbox on MATLAB Simulink Platform, the UPMs and UMPCs are established and the clocks (T_m) of them are set for $M = 10, 25, 100$. The control law data computed in the UMPCs are stored for up to each T_{free} in the "UMPCData" block and applied to the plant as open-loop instead the original MPC block in the Fig.4(a). Thus, the equations (66 and 77) are satisfied.

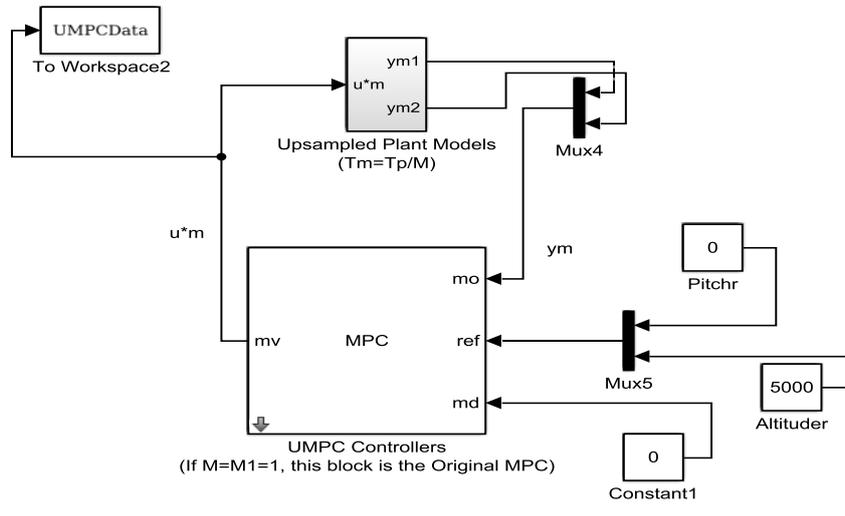
M depends on the clock speed of MATLAB software installed at Intel(R) Core(TM) i5-3210M microprocessor with 2.50 GHz and 4 GB RAM. The QPs (66 and 77) are run using MATLAB KWIK algorithm. Note that UMPC controllers with higher M can be simulated using MATLAB [27].

For the horizon $N = 10$, the matrices $q_w = I$, $r_w = 10$, the pitch reference value 0 rad. , the altitude reference value 5000 m. and M in the MPC and UMPC blocks, the responses and optimal control laws of them are illustrated in Fig. 5.

It proves that UMPCs settle faster than the original MPC to the reference values and there exist the free sampling periods $T_{free} = 0.18, 0.192$ and 0.198 sec. between them and when an improvement for the control is not made, the free time periods mean the energy efficiency for the UMPCs according to the original MPC.

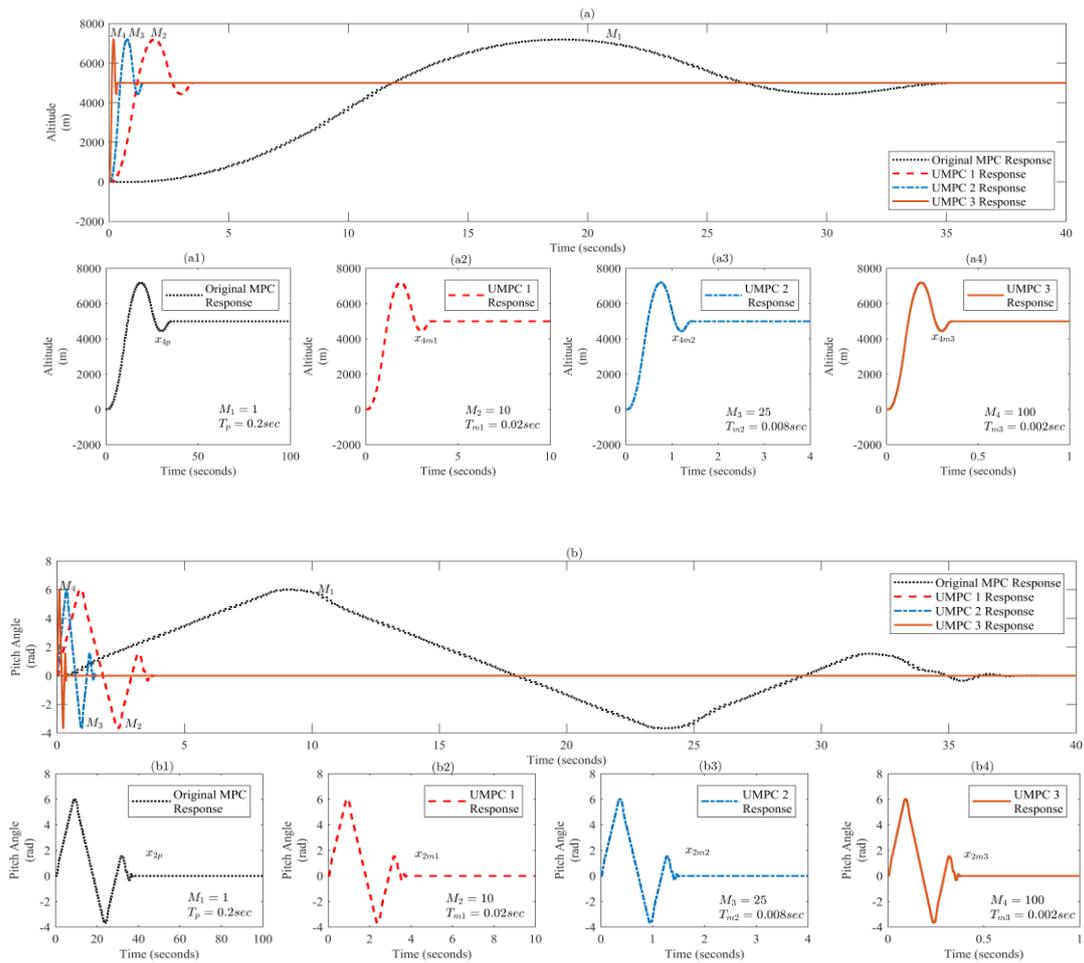


(a)



(b)

Fig.4 The Establishment on Simulink for the Control of Cessna Aircraft (a) Original MPC System ($M = 1$) and the Implementation of the Control Law Data computed in the UMPC Systems (b) UMPC Systems ($M = 10, 25, 100$)



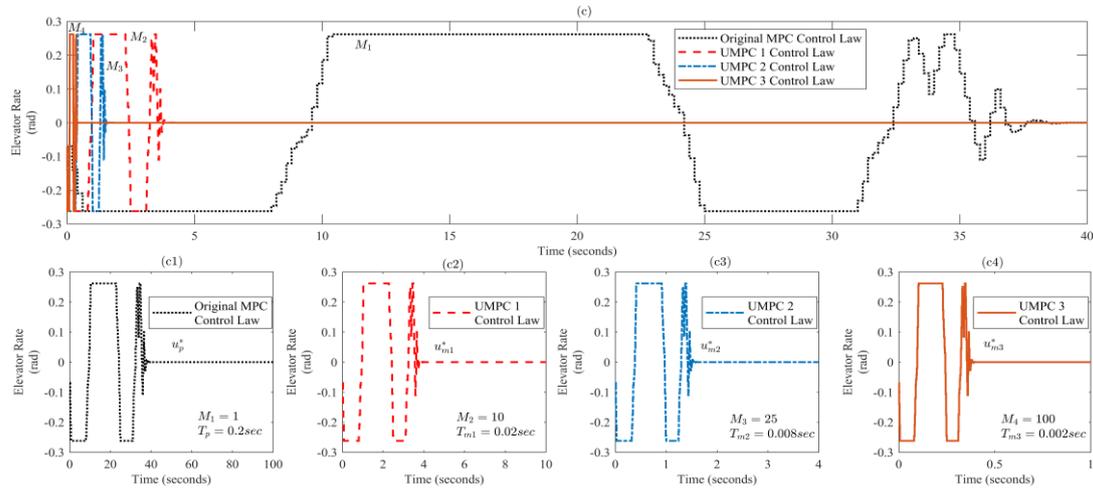


Fig.5 The Control Laws and Responses of Original MPC and UMPC for $M = 1, 10, 25, 100$ (a), The Altitude Responses of MPC and UMPC (b), The Pitch Responses of MPC and UMPC (c), The Control Laws of MPC and UMPC

Table 1: Original MPC and UMPC Responses

Controller	Scaling Factor M	Number of Iterations	Pitch / Altitude Rise Time (sec)	Pitch / Altitude Settling Time (sec)
Original MPC	1	170	1.810 / 6.813	34
UMPC 1	10	170	0.1810 / 0.6813	3.4
UMPC 2	25	170	0.0724 / 0.2722	1.36
UMPC 3	100	170	0.0181 / 0.0681	0.34

Table 2: Original MPC and UMPC Control Laws

Controller	Free Time Sampling Period (sec)	Number of Iterations	Total Free Time until Settling Time (sec)	Control Energy
Original MPC	0	170	0	1.9502
UMPC 1	0.18	170	30.6	0.1950
UMPC 2	0.192	170	32.64	0.0780
UMPC 3	0.198	170	33.66	0.0195

The performances of the controllers are compared in Table 1 and 2. For each M and 170 iterations, the Table 1 shows the rise and settling time responses of the controllers and Table 2 shows the free times and the energy values of the control signals. Energy value E of a signal $u(t)$ is computed by the following equation

$$E = \int_{-\infty}^{\infty} |u(t)|^2 dt \quad (84)$$

In the tables, adapting MPC algorithm (i.e. UMPC) by the selected scaling factors depending on the speed of the microprocessor, the controller energy and settling time decreases dramatically. As illustrated in Fig. 5, the first value of the computed optimal control law vector at kT_m for each UMPC controller is diluted by kT_{free} by saving its first value. Thus, the desired fast - response UMPC can be used instead of the original MPC.

Since the UMPC is compatible with the microprocessor and origin of speed of the UMPC is microprocessor, slow operation issue of MPC algorithms is solved for different time scaling factors by this method without reducing computation load of online optimization of used MPC algorithms. We used the standard MPC problem as the original MPC and compared it with the UMPC. The method can be applied to the methods in the References section and nonlinear dynamical systems, as well. For example, the method can be applied to Time Scaling Separation method in the reference [22]. For study, a high-scaled plant is separated into slow and fast dynamics. These dynamics are converted to slow and fast UPMs by scaling and sampling using a time scaling factor. Then, UMPCs are obtained based on UPMs. The UMPCs settle to desired output faster than the original slow and fast MPCs and have free times and lower control energies.

5. Conclusion

In this article, we introduce the UMPC method for the LTI dynamical systems. In the method, the original MPC strategy is contracted by using a time scaling factor based on the speed of the microprocessor in time domain. The original MPC is transformed to the UMPC. Therefore, the QPs of the original MPC and UMPC is the same except the online optimization sampling periods. The results prove that UMPC is settled to the reference values faster than the original MPC and provides lower control energy. In addition, since there exist free time periods between the original MPC and UMPC online optimization sampling periods, the first values of computed control law vectors in the UMPC are dilated to be able to equate it to the optimization sampling periods of the original MPC by the free time periods and then are applied to the plant at the original optimization sampling periods.

In addition, in free times, the UMPC algorithm can be improved by

- compensating modelling errors and disturbances by online system identification,
- increasing control capabilities using different control scenarios (such as optimization computations on varying horizon,
- incorporating with other control strategies (such as adaptive control and artificial intelligence algorithms).

6. Acknowledgements

The authors would like to thank to Dr. Serkan Gunel from Department of Electrical and Electronics Engineering of Dokuz Eylul University and Dr. Cuneyt Guzelis from Department of Electrical and Electronics Engineering of Yasar Eylul University for their valuable comments to improve the paper.

References

- [1.] Garcia CE, Prett DM, Morari M, 1989. Model predictive control: theory and practice - a survey. *Automatica*, 25(3):335-348.
- [2.] Van Den Boom TJ, Backx T, 2007. In: Model Predictive Control: College SC4060; Course Notes. TU Delft.
- [3.] Kouvaritakis B, Cannon M, 2016. In: Model Predictive control. Springer.
- [4.] Jerez J, Ling KV, Constantinides GA, et al., 2011. Mpc for deeply pipelined fpga implementation: Algorithms and circuitry. *IET Control Theory and Applications*, 8:1029-1041.
- [5.] Ling K, Yue S, Maciejowski J, 2006. A fpga implementation of model predictive control. IEEE / 2006 American Control Conference, Milwaukee, USA, p.1-7.
- [6.] Ling K.V., Wu B.F., J. Maciejowski, 2008. Embedded model predictive control (mpc) using a fpga. *Elsevier/IFAC Proceedings Volumes*, 41(2):15250-15255.
- [7.] Borrelli F, Bemporad A, Morari M, 2017. In: Predictive control for linear and hybrid systems. Cambridge University Press.
- [8.] Tøndel P, Johansen TA, Bemporad A, 2003. An algorithm for multi-parametric quadratic programming and explicit mpc solutions. *Automatica*, 39(3):489-497.
- [9.] Rao CV, Wright SJ, Rawlings JB, 1998. Application of interior-point methods to model predictive control. *Journal of optimization theory and applications*, 99(3):723-757.
- [10.] Bemporad A, Morari M, Dua V, et al., 2003. The explicit linear quadratic regulator for constrained systems. *Automatica*, 39(1):1-16.
- [11.] Fletcher R, 2013. Practical methods of optimization. In: . John Wiley & Sons.

- [12.] Cai X, Tippet MJ, Xie L, et al., 2014. Fast distributed mpc based on active set method. *Computers & Chemical Engineering*, 71:158-170.
- [13.] Borrelli F, Baotic' M, Pekar J, et al., 2010. On the computation of linear model predictive control laws. *Automatica*, 46(6):1035-1041.
- [14.] Wang Y, Boyd S, 2010. Fast model predictive control using online optimization. *IEEE Transactions on control systems technology*, 18(2):267-278.
- [15.] Yildirim EA, Wright SJ, 2002. Warm-start strategies in interior-point methods for linear programming. *SIAM Journal on Optimization*, 12(3):782-810.
- [16.] Koehler S, Danielson C, Borrelli F, 2017. A primal-dual active-set method for distributed model predictive control. *Optimal Control Applications and Methods*, 38(3):399-419.
- [17.] Morari M, 2013. Fast Model Predictive Control (MPC). In: Lecture Notes. Automatic Control Laboratory, ETH Zurich.
- [18.] Zeilinger, Nicole M, 2011. Real-time model predictive control. PHD Thesis, University of Stuttgart, Stuttgart.
- [19.] Liu Z, Xie L, Bemporad A, et al., 2018. Fast linear parameter varying model predictive control of buck dc-dc converters based on fpga. *IEEE*, 6:52434-52446.
- [20.] Qi R, Mei H, Chen C, et al., 2015. A fast mpc algorithm for reducing computation burden of mimo. *Chinese Journal of Chemical Engineering*, 23(12):2087-2091.
- [21.] Dunlap DD, Collins Jr EG, Caldwell CV, 2008. Sampling based model predictive control with application to autonomous vehicle guidance. Florida Conference on Recent Advances in Robotics, Denver,USA, p.1-6.
- [22.] Zhang Y, Nguyen H, Naidu DS, et al., 2014. Time scale analysis and synthesis for model predictive control. Proceedings of the WSEAS-NAUN 4th International Conference on Circuits, Systems, Control, Signals (CSCS 13), 9:1-6.
- [23.] Chen C, 1999. In: Linear system theory and design. Oxford University Press.
- [24.] Wolfram Alpha LCC, 2020. Software. <http://www.wolframalpha.com>
- [25.] Cannon M, 2016. C21 Model Predictive Control. In: Lecture Notes. Oxford University.
- [26.] Seborg DE, Mellichamp DA, Edgar TF, et al., 2010. In: Process dynamics and control. John Wiley & Sons.
- [27.] MathWorks, Inc., 2017. Software. <http://www.mathworks.com>