# SIMPLIFIED PROGRAMMING ALGORITHM FOR UNCONSTRAINED STATE SPACE MPC WITH STATE ESTIMATOR

Mohammed Tajudeen Jimoh[1], Ado Dan-Isa[2]
[1]Department of Mechanical Engineering, Bayero University, Kano, Nigeria.
[2]Department of Electrical Engineering, Bayero University, Kano, Nigeria.
*mtjimoh.mec@buk.edu,ng

**Keywords: –**
Model predictive control,
Optimal Control law,
Output prediction equation,
State estimator,
State space

**Article History: –**
Received: July, 2020.
Reviewed: August, 2020
Accepted: September, 2020
Published: September, 2020

## ABSTRACT

*This paper develops a simplified simulation algorithm unconstrained state space MPC that incorporates a state estimator for control of multivariable systems. A state space algorithm based on the augmented states with five tuning parameters (prediction and control horizon, output and input weights, and output filters), are presented. A block diagram showing the simulation plan is presented, together with algorithms for calculating the constant matrices of the MPC. Using Matlab and Matlab Simulink, the developed simulation plan is implemented on continuous models of two plants, a SISO system and a MIMO system. The implementation is simple and straight forward, presenting a very transparent state space MPC alternative for use by researchers.*

## 1. INTRODUCTION

Model Predictive Control (MPC) is an advanced control method that is being employed by many industries. Its formulation makes it very suitable for multivariable processes. Also, a large collection of modern control theory and analysis method can be applied easily to its algorithm, robustness, stability and development. MPC is a receding horizon control based on this principle: that at a sampling time$k$, information about measured plant output$y_k$is used to calculate a sequence of $M$ optimal control moves $\left[u_{(k)}, u_{(k+1)}, \dots, u_{(k+M-1)}\right]$ that ensures that a sequence of $P$ predicted future outputs $\left[\hat{y}_{(k+1)}, \hat{y}_{(k+2)}, \dots, \hat{y}_{(k+P)}\right]$ track a defined set-point (or target) $s_k$optimally. For the sequence of optimum $M$ control moves calculated at a sampling time $k$, only the first control move $u_k$ is sent to the plant as the manipulated variable. Then the same procedure is repeated at the next sampling time which is usually at regular sampling interval $t_s$ [1]-[6]. In MPC, $P$ is the prediction horizon and $M$ is the control horizon, and both are important MPC tuning parameters. There are additional three tuning parameters namely, the output weight $w$, the input weight $r$, and an output filter $\tau$. The output weight $w \geq 0$ is used to create controlled tracking of the output (or preferential tracking when there is more than one output according to their importance). A high $w$ usually results in faster tracking. Likewise, the input weight $r > 0$ is used to suppress the aggressive behaviour of the manipulated input (or to preferentially suppress some inputs when there is more than one input). Usually, the higher the value of $r$ on an input the more sluggish the control. The output filter $\tau$ ensures that a predicted output sequence makes a gradual transition from the current plant output measurement $y_k$ to the defined set-point sequence $s_k$ within the prediction horizon. We thus have five tuning parameters in MPC implementation namely $P$, $M$, $w$, $r$ and $\tau$.

State space Model Predictive Control (MPC) is so called because its formulation is based on the general state space equation. Generally, state space MPC (and other types of MPC) can be constrained (that is the formulation includes provisions for handling constraints on the inputs, the inputs changes, and the outputs), or unconstrained. Constrained MPC incorporates special optimisation quadratic programming technique for obtaining optimal control action. Unconstrained MPC on the other hand employs only simple linear programming to obtain optimal control actions. State space MPC formulations are well documented. See for instance [7]-[12].

Despite the existence of these publications and the fact that the principles underpinning MPC in general, and state space MPC in particular, is as simple as outlined above, writing computer programme for its implementation in the control of multivariable systems can be quite challenging for some researchers. This is especially so when deep insights are required into the effect of tuning parameters on the control actions, or on the robustness and stability analysis, on the analysis of numerical conditioning of its matrices, and on the need to reduce computational load. In addition, state space MPC requires a state estimator for improved control actions and the need to highlight this link cannot be over emphasised. Many researchers have to rely only on proprietary MPC software where their workings are sometimes opaque to the users. Because of this apparent disadvantage, this paper is focused on developing a simplified unconstrained, State space MPC algorithm for use in the control of multivariable systems that will be easy to programme using many programming languages.

The rest of the paper is organized as follows. In the next section, the formulation of unconstrained state space MPC using augmented state space models will be presented, followed by state estimation in Section 3. Then a matrix-based algorithm for the MPC and the state observer will be presented in Section 5, while examples utilizing the proposed algorithm will be presented in Section 5. The paper ends with conclusion and in recommendation in Section 6

## 2. MPC OBJECTIVE FUNCTION, PREDICTION EQUATION AND CONTROL LAW

This Section gives a formulation of State space MPC, incorporating the five tuning parameters mention in Section 1. The objective function, output prediction equation and the optimal input law which, constitute the basis of MPC, are presented here for a generalised Multiple-Input-Multiple-Output (MIMO) system. The formulations are presented with careful attention to the dimensions of the associated matrices and vectors, which would be vital to the formulation of the simplified MPC algorithm in Section 3

### 2.1 Objective Function

For a generalised MIMO system of $m$ outputs, $q$ inputs in which the weights on the output channels are $[w_1, w_2, \dots, w_\mathrm{m}]$, the input weights are $[r_1, r_2, \dots, r_\mathrm{q}]$, and the output filters are $[\tau_1, \tau_2, \dots, \tau_\mathrm{m}]$, the objective function may be written in matrix form as [12]:

$$\min_{\Delta \boldsymbol{U}} \boldsymbol{J} = (\boldsymbol{T} - \widehat{\boldsymbol{Y}})^T \boldsymbol{W} (\boldsymbol{T} - \widehat{\boldsymbol{Y}}) + (\Delta \boldsymbol{U})^T \boldsymbol{R} (\Delta \boldsymbol{U}) \qquad (1)$$

where $\boldsymbol{W} \in \mathbb{R}^{mP \times mP}$ and $\boldsymbol{R} \in \mathbb{R}^{qM \times qM}$ are square matrices of output and input weights elements respectively. These are defined as follows:

$$\boldsymbol{W} = \begin{bmatrix} W_m & 0 & \cdots & 0 \\ 0 & W_m & 0 & 0 \\ \vdots & 0 & \ddots & \vdots \\ 0 & 0 & \cdots & W_m \end{bmatrix} \qquad (2)$$

$$\boldsymbol{R} = \begin{bmatrix} R_q & 0 & \cdots & 0 \\ 0 & R_q & 0 & 0 \\ \vdots & 0 & \ddots & \vdots \\ 0 & 0 & \cdots & R_q \end{bmatrix} \qquad (3)$$

where $W_m$ and $R_q$ are diagonal matrices whose elements are the outputs and inputs weights respectively and they are defined as:

$$W_m = \begin{bmatrix} w_1 & 0 & \cdots & 0 \\ 0 & w_2 & 0 & 0 \\ \vdots & 0 & \ddots & \vdots \\ 0 & 0 & \cdots & w_m \end{bmatrix} \qquad (4)$$

$$R_q = \begin{bmatrix} r_1 & 0 & \cdots & 0 \\ 0 & r_2 & 0 & 0 \\ \vdots & 0 & \ddots & \vdots \\ 0 & 0 & \cdots & r_q \end{bmatrix} \qquad (5)$$

The reference trajectory $\boldsymbol{T} \in \mathbb{R}^{mP \times 1}$is a column vector defined as:

$$\boldsymbol{T} = [T_{y,1}, T_{y,2}, T_{y,3}, \dots, T_{y,P}]^T + [T_{s,1}, T_{s,2}, T_{s,3}, \dots, T_{s,P}]^T \qquad (6)$$

where

$$T_{y,i} = [\tau_1^i y_{1k}, \tau_2^i y_{2k}, \dots, \tau_m^i y_{mk},], \quad i = 1, 2, \dots, P \qquad (7)$$

$$T_{s,i} = [(1 - \tau_1^i) s_{1,k}, (1 - \tau_2^i) s_{2,k}, \dots, (1 - \tau_m^i) s_{m,k}], \quad i = 1, 2, \dots, P \qquad (8)$$

Here, $\tau_j^i$ indicates $\tau$ for output channel $j$ raised to power $i$, and $s_{j,k}$ is the set point for output channel $j$ at step $k$. The derivation for the output prediction $\widehat{\boldsymbol{Y}}$ and its dimension are given in the following sub-section.

### 2.2 The Output Prediction Equation

For a proper MIMO system represented by states space equation of $n$ states, and in which the state vector $\boldsymbol{x}$, output vector $\boldsymbol{y}$ and input vector $\boldsymbol{u}$ have dimensions $\boldsymbol{x} \in \mathbb{R}^{n \times 1}$, $\boldsymbol{y} \in \mathbb{R}^{m \times 1}$and $\boldsymbol{u} \in \mathbb{R}^{q \times 1}$ respectively, and given that the system matrix $\boldsymbol{A}$, input matrix $\boldsymbol{B}$ and the output matrix $\boldsymbol{C}$ have dimensions $\boldsymbol{A} \in \mathbb{R}^{n \times n}$, $\boldsymbol{B} \in \mathbb{R}^{n \times q}$, $\boldsymbol{C} \in \mathbb{R}^{m \times n}$, the augmented state equation can be written as [12]:

$$\begin{bmatrix} \Delta x_{(k+1)} \\ y_{(k+1)} \end{bmatrix} = \begin{bmatrix} \boldsymbol{A} & 0_{n,m} \\ \boldsymbol{CA} & I_m \end{bmatrix} \begin{bmatrix} \Delta x_k \\ y_k \end{bmatrix} + \begin{bmatrix} \boldsymbol{B} \\ \boldsymbol{CB} \end{bmatrix} \Delta \boldsymbol{U_k} \qquad \text{(9a)}$$
$$y_{(k)} = [0_{n,m}^T \quad I_m] \boldsymbol{X_{g(k)}}$$

or simply as:

$$\boldsymbol{X_{g(k+1)}} = \boldsymbol{A_g X_{g(k)}} + \boldsymbol{B_g} \Delta \boldsymbol{U_{(k)}} \qquad \text{(9b)}$$
$$y_{(k)} = \boldsymbol{C_g X_{g(k)}}$$

where $\boldsymbol{A_g}, \boldsymbol{B_g}$and $\boldsymbol{C_g}$ are augmented matrices with the following dimensions: $\boldsymbol{A_g} \in \mathbb{R}^{(n+m) \times (n+m)}$, $\boldsymbol{B_g} \in \mathbb{R}^{(n+m) \times q}$, $\boldsymbol{C_g} \in \mathbb{R}^{m \times (n+m)}$. The new state matrix $\boldsymbol{X_g}$ has dimension $\boldsymbol{X_g} \in \mathbb{R}^{(n+m) \times 1}$, and the input change vector $\Delta \boldsymbol{U}$ has the dimension $\Delta \boldsymbol{U} \in \mathbb{R}^{q \times 1}$. $0_{n,m}$ is an all-zeros elements matrix of $n$ rows and $m$ columns, and $0_{n,m}^T$ is its transpose. $I_m$ is an identity matrix of dimension $m$.

Using the augmented state models of Equation (9), the output prediction equation within the prediction and control horizons $P$ and $M$for state space MPC can be written in generalised vector-matrix form as [12]:

$$\begin{bmatrix} \hat{Y}_{(k+1)} \\ \hat{Y}_{(k+2)} \\ \hat{Y}_{(k+3)} \\ \vdots \\ \hat{Y}_{(k+P)} \end{bmatrix} = \begin{bmatrix} \boldsymbol{C_g B_g} & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \boldsymbol{C_g A_g B_g} & \boldsymbol{C_g B_g} & \cdots & \boldsymbol{0} \\ \boldsymbol{C_g A_g^2 B_g} & \boldsymbol{C_g A_g B_g} & \cdots & \boldsymbol{0} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{C_g A_g^{P-1} B_g} & \boldsymbol{C_g A_g^{P-2} B_g} & \cdots & \boldsymbol{C_g A_g^{P-M} B_g} \end{bmatrix} \begin{bmatrix} \Delta \boldsymbol{U}_{(\boldsymbol{k})} \\ \Delta \boldsymbol{U}_{(\boldsymbol{k+1})} \\ \Delta \boldsymbol{U}_{(\boldsymbol{k+2})} \\ \vdots \\ \Delta \boldsymbol{U}_{(\boldsymbol{k+M-1})} \end{bmatrix} + \begin{bmatrix} \boldsymbol{C_g A} \\ \boldsymbol{C_g A_g^2} \\ \boldsymbol{C_g A_g^3} \\ \vdots \\ \boldsymbol{C_g A_g^P} \end{bmatrix} \boldsymbol{X_{g(k)}} \quad (10a)$$

or simply as:

$$\widehat{\boldsymbol{Y}} = \boldsymbol{\Phi} \Delta \boldsymbol{U} + \boldsymbol{\Gamma X_g} \quad (10b)$$

where the dimensions of $\boldsymbol{\Phi}$, $\boldsymbol{\Gamma}$, $\widehat{\boldsymbol{Y}}$ and $\Delta\boldsymbol{U}$ are $\boldsymbol{\Phi} \in \mathbb{R}^{mP\times qM}$,$\boldsymbol{\Gamma} \in \mathbb{R}^{mp\times(n+m)}$, $\widehat{\boldsymbol{Y}} \in \mathbb{R}^{mP\times 1}$and $\Delta\boldsymbol{U} \in \mathbb{R}^{qM\times 1}$.

$\hat{Y}_{(k+j)}$is defined as $\left[\hat{y}_{1,k+j}, \hat{y}_{2,k+j}, \hat{y}_{3,k+j}, \dots, \hat{y}_{m,k+j}\right]^T$ and $\Delta U_{(k+i)}$ is defined as $\left[\Delta u_{1,k+i}, \Delta u_{2,k+i}, \Delta u_{3,k+i}, \dots, \Delta u_{q,k+i}\right]^T$.

The following sub-section gives the derivation of the optimum control law for calculating $\Delta\boldsymbol{U}$.

### 2.3 The Optimal Control Law

For linear unconstrained MPC, the optimal control law is obtained by finding the least squares solution of the objective function of Equation (1). By substituting the prediction outputs Equation (10) into Equation (1), we have:

$$\min_{\Delta \boldsymbol{U}} J = -2\Delta\boldsymbol{U}^T\Phi^T\boldsymbol{WE} + \Delta\boldsymbol{U}^T(\boldsymbol{\Phi}^T\boldsymbol{W\Phi} + \boldsymbol{R})\Delta\boldsymbol{U} + \boldsymbol{E}^T\boldsymbol{E} \quad (11)$$

where $\boldsymbol{E}$ is given as:

$$\boldsymbol{E} = \boldsymbol{T} - \Gamma\boldsymbol{X_g} \quad (12)$$

The first derivative of the cost function $J$ gives:

$$\frac{\partial J}{\partial \Delta\boldsymbol{U}} = -2\boldsymbol{\Phi}^T\boldsymbol{WE} + 2(\boldsymbol{\Phi}^T\boldsymbol{W\Phi} + \boldsymbol{R})\Delta\boldsymbol{U} = 0 \quad (13)$$

Then the optimum control law for unconstrained MPC then becomes:

$$\Delta U = (\boldsymbol{\Phi}^T\boldsymbol{W\Phi} + \boldsymbol{R})^{-1}\boldsymbol{\Phi}^T\boldsymbol{WE} \quad (14a)$$

or as

$$\Delta\boldsymbol{U} = \boldsymbol{GE} = \boldsymbol{G}(\boldsymbol{T} - \Gamma\boldsymbol{X_g}) \quad (14b)$$

where

$$\boldsymbol{G} = (\boldsymbol{\Phi}^T\boldsymbol{W\Phi} + \boldsymbol{R})^{-1}\boldsymbol{\Phi}^T\boldsymbol{W} \quad (15)$$

It can be seen from the optimal control law of Equation (14) that for time-invariant system, $\boldsymbol{G}$ remains constant throughout and $\boldsymbol{T}$ is a vector which depends on outputs measurements and set points. The accuracy of the calculation of optimal control input $\Delta\boldsymbol{U}$ for unconstrained case therefore depends very heavily on the prediction of the augmented state vector $\boldsymbol{X_g}$ at every sampling instant. For deadbeat control (where the there is no model-plant mismatch), this equation is sufficient. But in reality, there is always model-plant mismatch for reasons such as the use of simplified process model, the use of linearized model for a non-linear or unstable plant. Also, for real plant, the process states are mostly not measurable or accessible. For these reasons, state space MPC usually incorporates a state estimator or an *observer*. Even where the states are measurable, incorporating a state estimator in a process may serve as soft sensors, thereby serving as practical or economical alternative to actual measurement.

## 3. STATE ESTIMATION EQUATIONS

Assuming that the process is stochastic, where the process is excited by random white noise and the measurement contain random noise, such that the states equations are written as [10], [12], [15]:

$$\begin{aligned} \boldsymbol{X_{g(k+1)}} &= \boldsymbol{A_g X_{g(k)}} + \boldsymbol{B_g}\Delta\boldsymbol{U}_{(k)} + \boldsymbol{FH} \\ y_{(k)} &= \boldsymbol{C_g X_{g(k)}} + \boldsymbol{N} \end{aligned} \quad (16)$$

where $\boldsymbol{H} \in \mathbb{R}^{(n+m)\times(n+m)}$ is the auto-covariance diagonal matrix of the process white noise and $\boldsymbol{N} \in \mathbb{R}^{m\times m}$ is the auto-covariance diagonal matrix of the measurement white noise. The matrix $\boldsymbol{F} \in \mathbb{R}^{(n+m)\times(n+m)}$ is usually an identity matrix.

Using the discrete-time predictor-corrector version of the discrete-time Kalman Filter, we can write the improved predicted state for the next sampling time as [10], [14]:

$$\widehat{\boldsymbol{X}}_{\boldsymbol{g}(k+1)} = \left[\widehat{\boldsymbol{X}}_{\boldsymbol{g(k)}} + \boldsymbol{K}\left(\boldsymbol{Y}_{(k)} - \boldsymbol{C_g}\widehat{\boldsymbol{X}}_{\boldsymbol{g(k)}}\right)\right]\boldsymbol{A_g} + \boldsymbol{B_g}\Delta\boldsymbol{U}_{(\boldsymbol{k})} \quad (17)$$

The steady state gain $\boldsymbol{K} \in \mathbb{R}^{(n+m)\times m}$ is calculated by solving the following Kalman Filter equations iteratively [15]:

$$K_{(k)} = P_{p(k)}\boldsymbol{C_g^T}\left[C_g P_{p(k)}\boldsymbol{C_g^T} + N\right]^{-1} \quad (18)$$

$$P_{c(k)} = \left[I - K_{(k)}\boldsymbol{C_g}\right]P_{p(k)} \quad (19)$$

$$P_{p(k+1)} = \boldsymbol{A_g}P_{c(k)}\boldsymbol{A_g^T} + FHF^T \quad (20)$$

where:

$\boldsymbol{K}_{(k)}$ is the Kalman gain at $k$th iterate

$P_{p(k)}$ ($P_p \in \mathbb{R}^{(n+m)\times(n+m)}$) is the auto-covariance matrix of the estimation error of the predicted ($\hat{X}_{g(k)}$) state estimate.

$P_{c(k)}$ ($P_c \in \mathbb{R}^{(n+m)\times(n+m)}$) is the auto-covariance matrix of the estimation error of the corrected predicted ($\hat{X}_{gc(k)}$) state estimate.

For linear, time-invariant model, the matrices $P_p$ and $P_c$ each converge towards steady-state values, for an initial guess $P_{p(0)}$. The value of $K_{(k)}$ at this steady state is the state estimator gain used in Equation (14).

The error model for the corrected stated estimate is given by:

$$e_{xc(k+1)} = \left(I - KC_g\right)A_g e_{xc(k)} + \left(I - KC_g\right)Gv_{(k)} - Kw_{(k+1)} \tag{21}$$

In other words, the state gain matrix $\boldsymbol{K}$ that makes $e_{xc(k+1)} \to 0$ as $k \to \infty$ in Equation (21) is used in the state estimation Equation (13). Apart from ensuring that $\hat{X}_{g(k+1)} \to 0$ as $k \to \infty$, $K$ must also be chosen such that the observer dynamics is much faster than the system

## 4. ALGORITHMS FOR CALCULATING MPC MATRICES AND VECTORS FOR MIMO SYSTEMS

From optimal control law of Equation (14), it is obvious that while the matrix $\boldsymbol{G}$ is a constant and can be obtained once, the vector $\boldsymbol{E}$, and by extension the vectors $\boldsymbol{T}$ and $\boldsymbol{X_g}$, have to be computed at each sampling instant. So, for a MIMO MPC implementation, the major problem concerns the ability to obtain these vectors at the appropriate time during the entire prediction period. To simplify the procedure for obtaining the matrices and vector, we proceed as follows:

Let the reference trajectory vector $\boldsymbol{T}$ for the entire prediction horizon $P$ be written generally as:

$$\boldsymbol{T} = \boldsymbol{\alpha Y} + \boldsymbol{\beta S} \tag{22}$$

where $\boldsymbol{Y}$ and $\boldsymbol{S}$ are vectors of actual plant outputs and set-points within the prediction horizon respectively. Both $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are constant matrices.

Then, the optimum control law may be written as:

$$\Delta\boldsymbol{U} = \left(\boldsymbol{G\alpha Y} + G\boldsymbol{\beta S} - \boldsymbol{G}\Gamma\hat{\boldsymbol{X}}_{\boldsymbol{g}}\right) \tag{23a}$$

or as:

$$\Delta\boldsymbol{U} = \boldsymbol{K_y Y} + \boldsymbol{K_s S} - \boldsymbol{K_x}\hat{\boldsymbol{X}}_{\boldsymbol{g}} \tag{23b}$$

where

$$\boldsymbol{K_y} = \boldsymbol{G\alpha} \tag{24}$$

$$\boldsymbol{K_s} = \boldsymbol{G\beta} \tag{25}$$

$$\boldsymbol{K_x} = \boldsymbol{G}\Gamma \tag{26}$$

For MPC as receding horizon control where only the first $q$ inputs ($\Delta\boldsymbol{U_k}$) are sent to the plant, we can write

$$\Delta\boldsymbol{U_k} = \boldsymbol{K_u}\Delta\boldsymbol{U} \tag{27}$$

Equations (23) and (24) indicate that $\boldsymbol{K_y}$, $\boldsymbol{K_s}$, $\boldsymbol{K_x}$ and $\boldsymbol{K_u}$may be calculated as constant matrices at the beginning of MPC implementation and the problem reduces to estimating the vectors $\boldsymbol{Y}$, $\boldsymbol{S}$ and $\hat{\boldsymbol{X}}_{\boldsymbol{g}}$at every sampling time. To obtain the constant vectors and matrices required to compute the optimum control input at every sampling time, we proceed as follows:

a) Define the following constant vectors and matrices

   $\boldsymbol{V_P}$ as a column vector with $P$ ones, i.e. $V_P \in \mathbb{R}^{P\times 1}$

   $\boldsymbol{I_P}$ as an identity matrix of $P$ rows and $P$ columns,. i.e. $\boldsymbol{I_P} \in \mathbb{R}^{P\times P}$

   $\boldsymbol{I_M}$ as an identity matrix of $m$ rows and $m$ columns, i.e. $\boldsymbol{I_M} \in \mathbb{R}^{M\times M}$

   $\boldsymbol{I_q}$ as an identity matrix of $q$ rows and $q$ columns, i.e. $\boldsymbol{I_q} \in \mathbb{R}^{q\times q}$

   $\boldsymbol{I_{mP}}$ as an identity matrix of $mP$ rows and $mP$ columns, i.e. $I_{mP} \in \mathbb{R}^{mP\times mP}$

   $\boldsymbol{0_{q,z}}$ as a matrix of zeros with $q$ rows and $z = (M-1)q$ columns, i.e. $0_{q,z} \in \mathbb{R}^{q\times(M-1)q}$

b) Define the augmented state matrices as given in Equations (9) and the matrices $\boldsymbol{\Phi}$ and $\boldsymbol{\Gamma}$of Equation (10)

c) Define the weights on the output and input channels respectively as diagonal matrices as shown in Equations (2) and (3). Then output weight diagonal matrices $\boldsymbol{W} \in \mathbb{R}^{mP\times mP}$ is defined as the Kronecker product of identity matrix $I_P$ and the matrix $W_m$ of Equation (4) as:

$$\boldsymbol{W} = \boldsymbol{I_P} \otimes W_m \tag{28}$$

   The input weight diagonal matrices $\boldsymbol{R} \in \mathbb{R}^{qM\times qM}$ is defined as:

$$\boldsymbol{R} = \boldsymbol{I_M} \otimes R_q \tag{29}$$

d) Let the filters $\tau_1, \tau_2, \ldots, \tau_m$, on the output channels be the elements of a diagonal matrix $\alpha_i$ $(i = 1, 2, \ldots, P)$ such that:

$$\alpha_i = \begin{bmatrix} \tau_1^i & 0 & \cdots & 0 \\ 0 & \tau_2^i & \vdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \tau_m^i \end{bmatrix} \tag{30}$$

Then constant matrices $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ of Equation (28) are defined as follows (see Equation 34):

$$\boldsymbol{\alpha} = \begin{bmatrix} \alpha_1 & 0 & \cdots & 0 \\ 0 & \alpha_2 & \vdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \alpha_P \end{bmatrix} \tag{31}$$

$$\boldsymbol{\beta} = \boldsymbol{I_{mP}} - \boldsymbol{\alpha} \tag{32}$$

e) Then define the constant matrices $\boldsymbol{G}$,$\boldsymbol{K_y}$, $\boldsymbol{K_s}$ and$\boldsymbol{K_x}$as shown in Equation (23)

At every sampling time $k$, let the column vectors for measure plant output $y_k$ and set-point $S_k$for all output channels be defined as:

$$Y_k = [y_1, y_2, \ldots, y_m]^T \tag{33}$$

$$S_k = [s_1, s_2, \ldots, s_m]^T \tag{34}$$

Then at every sampling instant $k$, the following are defined:

f) The vector $\boldsymbol{Y}$ of current plant output measurements for the prediction horizon $P$ is defined is the Kronecker product of column the vector $\boldsymbol{V_P}$ and the column vector $Y_k$ as:

$$\boldsymbol{Y} = \boldsymbol{V_P} \otimes Y_k \tag{35}$$

g) The vector $\boldsymbol{S}$ of set points for the prediction horizon $P$ is defined is the Kronecker product of column vector $\boldsymbol{V_P}$ and the column vector $S_k$ as:

$$\boldsymbol{S} = V_P \otimes S_k \tag{36}$$

h) Finally, define the matrix $\boldsymbol{K_x}$ as

$$\boldsymbol{K_x} = [\boldsymbol{I_q}, 0_{q,z}] \tag{37}$$

The block diagram for unconstrained state space MPC control incorporating state estimator and with all the constant matrices defined above is shown in Figure (1).

In Figure 1, the two dotted blocks after $S_k$and $Y_k$ indicate that the mathematical Kronecker products $\boldsymbol{V_P} \otimes S_k$ and $\boldsymbol{V_P} \otimes Y_k$ must be obtained.

## 5. APPLIED EXAMPLES

We demonstrate the use of the simplified simulation plan with two examples. The first example is the transfer function of a SISO system representing the model of a helicopter in a particular flight condition given by [10]:

$$G(s) = \frac{u(s)}{y(s)} = \frac{9.8s^2 - 4.9s + 61.74}{s^3 + 0.4199s^2 - 0.006028s + 0.09802} \tag{38}$$

where $u$ is the helicopter's rotor angle, and $y$ is its forward speed.

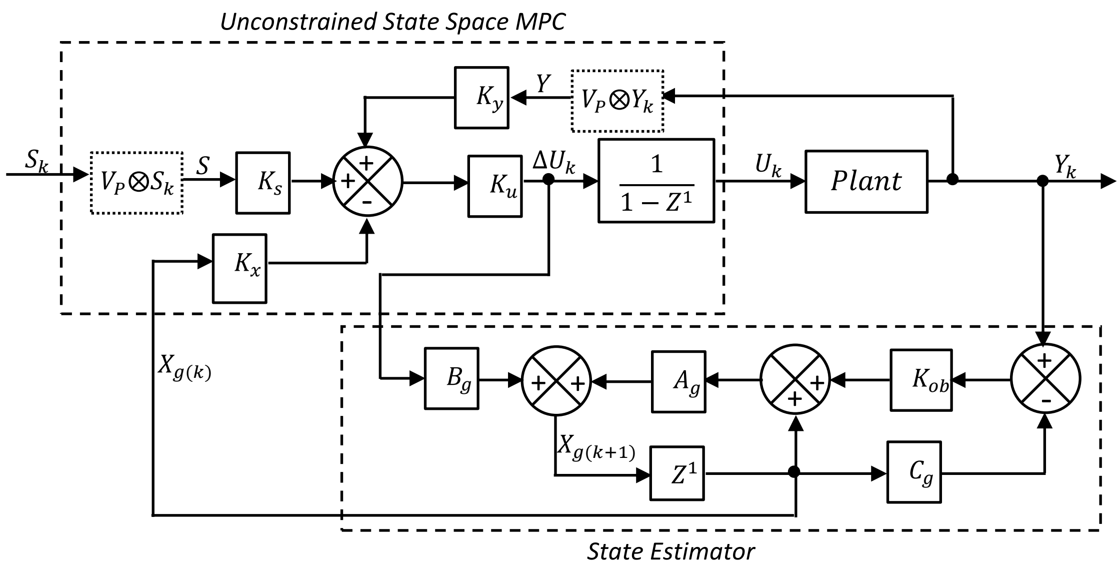


Fig. 1: Block diagram of unconstrained MPC with state estimator

The second example is the linearized continuous state space model of a MIMO system representing the dynamics of a paper machine headbox also given by [10]:

$$A = \begin{bmatrix} -1.93 & 0 & 0 & 0 \\ 0.394 & -0.426 & 0 & 0 \\ 0 & 0 & -0.63 & 0 \\ 0.82 & -0.784 & 0.413 & -0.426 \end{bmatrix}$$
$$B = \begin{bmatrix} 1.274 & 1.274 \\ 0 & 0 \\ 1.34 & -0.65 \\ 0 & 0 \end{bmatrix} \qquad (39)$$
$$C = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad D = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

The two inputs are $u1$ (stock flowrate) and $u2$ (WW flowrate), while the three outputs are $y1$ (Headbox level) $y2$ (Feed tank consistency) and $y3$ (Headbox consistency)

The parameters of the state space MPC implemented on the systems are given in Table 1, while the values of the calculated constant matrices of Figure 1 are given in Tables (2) and (3) for the Helicopter model and Paper Machine model respectively.

The parameters of the state space MPC implemented on the systems are given in Table 1, while the values of the calculated constant matrices of Figure 1 are given in Tables (2) and (3) for the Helicopter model and Paper Machine model respectively.

The matrices of Tables 2 and 3 are calculated using Matlab while the State Space MPC simulation are implemented using Matlab Simulink (the entire simulation can also be carried out using only Matlab, without Simulink). While the MPC and the state estimator are in discrete states, the plants are implemented as continuous systems (as presented in Equations (37) and (38)), closer to real situation. The plots of the responses from the MPC simulations are shown in Figures 2 to 4.

Since in the Paper Machine example, the system is non-square, with more controlled variables than manipulated variable, the input weight of one of the outputs ($y2$) was set to zero. This explains why only $y2$ does not track its set-point. All other outputs track their set-points as shown in Figures 2 and 3.

.

Table 1: MPC Parameters

| **Parameter** | **Helicopter** | **Paper Machine** |
|---|---|---|
| Sampling Time ($T_s$) | 0.6 | 1 |
| Prediction Horizon ($P$) | 5 | 5 |
| Control Horizon ($M$) | 2 | 2 |
| Input weights ($Rm$) | 1 | [0.6 0.6] |
| Output weights ($Wm$) | 1 | [10 0 10] |
| Output Filters ($\tau$) | 0.9 | [0.9 0.9 0.9] |
| Covariance matrix (QQ) | [1, 0, 0, 0<br>0, 1, 0, 0<br>0, 0, 1, 0<br>0, 0, 0, 1] | [1, 0, 0, 0, 0, 0, 0<br>0, 1, 0, 0, 0, 0, 0<br>0, 0, 1, 0, 0, 0, 0, 0<br>0, 0, 0, 1, 0, 0, 0<br>0, 0, 0, 0, 1, 0, 0<br>0, 0, 0, 0, 0, 1, 0<br>0, 0, 0, 0, 0, 0, 1] |
| Covariance Matrix (RR) | 1 | [0.1, 0, 0<br>0, 0.1, 0<br>0, 0, 0.1] |

Table 2: Constant Simulation matrices for MPC control of Helicopter

| **Matrices** | **Calculated Constant** |
|---|---|
| $A_g$ | [0.7751, -0.0131, -0.0520, 0<br>0.5301, 0.9977, -0.0162, 0<br>0.1658, 0.5997, 0.9967, 0<br>15.2325, 32.0089, 61.1056, 1.0000] |
| $B_g$ | [0.5301; 0.1658; 0.0338; 6.4720] |
| $C_g$ | [0, 0, 0, 1] |
| $K_x$ | [2.8179, 6.2009, 7.5510, 0.0574<br>-1.2796, -6.6538, -11.1552, -0.0948] |
| $K_y$ | [0.0098, 0.0135, 0.0186, 0.0129, -0.0090<br>-0.0181, -0.0247, -0.0336, -0.0217, 0.0207] |
| $K_s$ | [0.0011, 0.0032, 0.0069, 0.0067, -0.0063<br>-0.0020, -0.0058, -0.0125, -0.0114, 0.0144] |
| $K_u$ | [1, 0, 0] |
| $K_{ob}$ | [0.0043; 0.0172; 0.0169; 1.0000] |

Table 3: Constant Simulation matrices for MPC control of Paper Machine Headbox

| | **Calculated Constant Matrices** |
|---|---|
| $A_g$ | [0.1451, 0, 0, 0, 0, 0, 0<br>0.1331, 0.6531, 0, 0, 0, 0, 0<br>0 0 0.5326 0 0 0 0<br>0.2122 -0.5120 0.2440 0.6531 0 0 0<br>0.1331 0.6531 0 0 1.0000 0 0<br>0 0 0.5326 0 0 1.0000 0<br>0.2122 -0.5120 0.2440 0.6531 0 0 1.0000] |
| $B_g$ | [0.5643 0.5643<br>0.1239 0.1239<br>0.9942 -0.4822<br>0.4199 0.1283<br>0.1239 0.1239<br>0.9942 -0.4822<br>0.4199 0.1283] |
| $C_g$ | [0, 0, 0, 0, 1, 0, 0<br>0, 0, 0, 0, 0, 1, 0<br>0, 0, 0, 0, 0, 0, 1] |
| $K_x$ | [0.2658 -1.9337 0.5959 1.3457 -0.3378 0 1.4922<br>0.4513 3.3857 -0.3531 -0.5514 2.2152 0 -0.2534<br>-0.0688 0.9715 -0.0561 -0.6030 0.6205 0 -1.1193<br>0.1112 0.9417 -0.1867 -0.2541 -0.3237 0 -0.3140] |
| $K_y$ | [0.1020, 0, 0.5341, 0.0512, 0, 0.5848, -0.0557, 0, 0.2692, -0.1201, 0, 0.0199, -0.1506, 0, -0.1316<br>0.4832, 0, 0.3119, 0.5455, 0, -0.1059, 0.3520, 0, -0.2817, 0.2085, 0, -0.1416, 0.1206, 0, 0.0785<br>-0.0811, 0, -0.6163, -0.0128, 0, -0.6742, 0.1010, 0, -0.2492, 0.1672, 0, 0.1095, 0.1966, 0, 0.3382<br>-0.5781, 0, -0.4745, -0.4679, 0, 0.1387, -0.0457, 0, 0.3659, 0.2245, 0, 0.0740, 0.3641, 0, -0.3382] |
| $K_s$ | [0.0113, 0, 0.0593, 0.0120, 0, 0.1372, -0.0207, 0, 0.1001, -0.0630, 0, 0.0104, -0.1044, 0, -0.0913<br>0.0537, 0, 0.0347, 0.1279, 0, -0.0249, 0.1309, 0, -0.1047, 0.1093, 0, -0.0742, 0.0836, 0, 0.0545<br>-0.0090, 0, -0.0685, -0.0030, 0, -0.1581, 0.0376, 0, -0.0926, 0.0876, 0, 0.0574, 0.1363, 0, 0.2345<br>-0.0642, 0, -0.0527, -0.1097, 0, 0.0325, -0.0170, 0, 0.1360, 0.1177, 0, 0.0388, 0.2525, 0, -0.2345] |
| $K_u$ | [1, 0, 0, 0<br>0, 1, 0, 0] |
| $K_{ob}$ | [0.0165, -0.0022, 0.0161<br>0.2982, 0.0204, -0.1346<br>0.0204, 0.2274, 0.0599<br>-0.1421, 0.0629, 0.4794<br>0.9419, 0.0019, -0.0120<br>0.0019, 0.9359, 0.0053<br>-0.0120, 0.0053, 0.9578] |

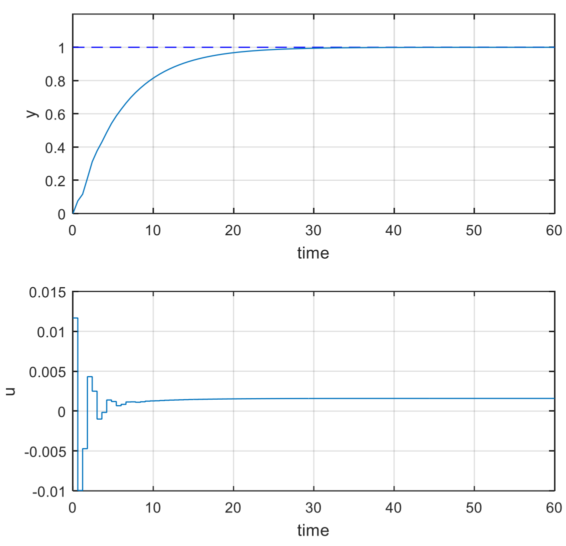


Fig. 2: Output and input trends of the Helicopter control

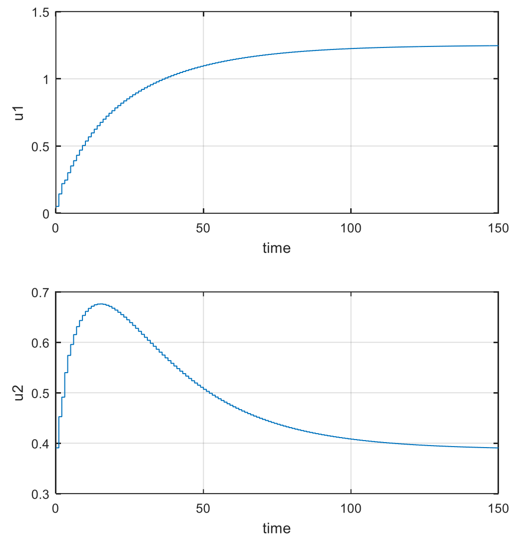


Fig. 3: Input trends of the Paper Machine control

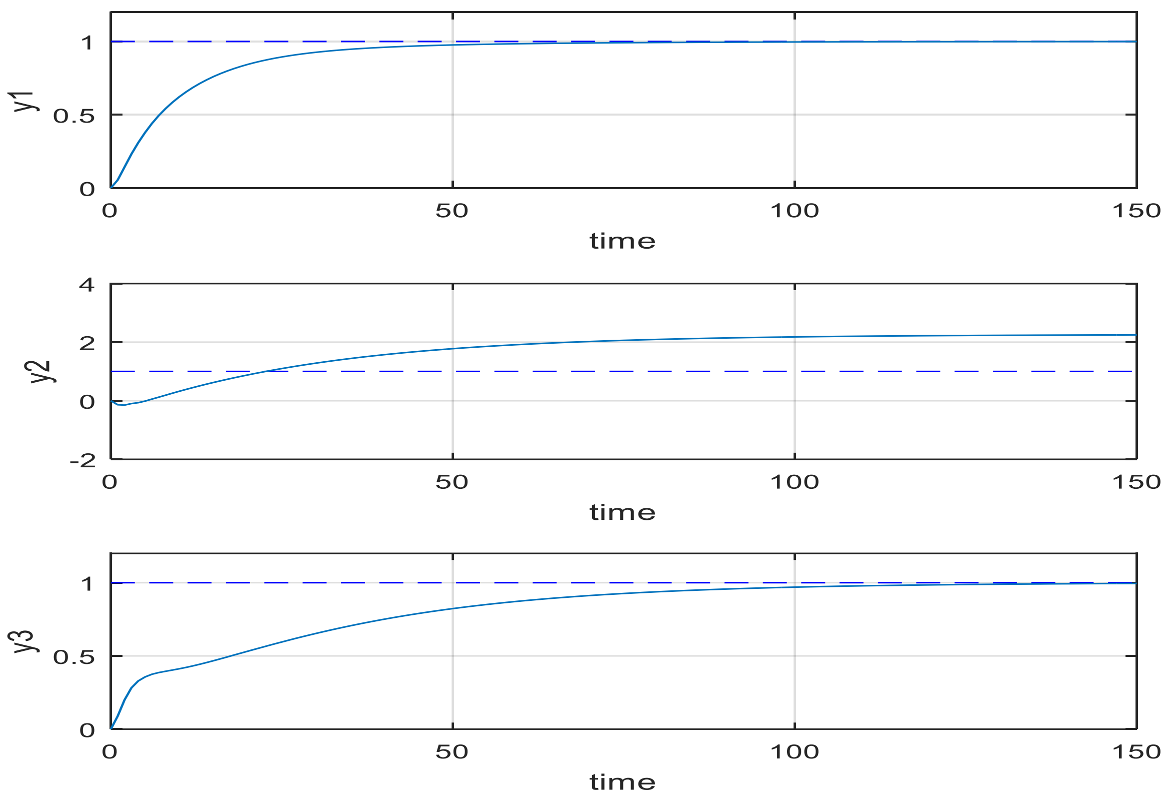


Fig. 4: Output trends of the Paper Machine control

## 6. CONCLUSION

The algorithms for a State Space MPC using augmented state models was presented and a simplified simulation plan for State Space MPC, incorporating state estimator, has been developed. The simulation plan is easy to implement. The MPC implementations, using models of a SISO system and a MIMO system indicate that the outputs track the set-points very well. Their implementations present very transparent state space MPC alternative for use by researchers.

## REFERENCES

[1] Camacho, E. F. & A., B. C., 2007 . *Model Predictive Control.* s.l.:Springer.

[2] Kouvaritakis, B. & Cannor, M., 2016. *Model Predictive Control.* s.l.:Springer.

[3] Rawlings, J. B., 2000. Tutorial overview of model predictive control. Control Systems. *IEEE*, Volume 20, pp. 38-52..

[4] Rossiter, J. A., 2018. *A first Course in Predictive Control.* 2nd ed. Boca Raon: CRC Press

[5] Rakovic, S. & Levine, W., 2018. *Handbook of Model Predictive Control.* Basel: Birkhauser.

[6] Seborg, D. E., Edgar, T. F., Mellichamp, D. A. & Doyle, F. J., 2016. *Process Dynamics and Control,.* 4th ed. s.l.:John Wiley & Sons .

[7] Bemporad, A., Ricker, N. L. & Morari, M., 2020. *Model Predictive Control Toolbox User's Guide.* s.l.:The Mathworks, Inc.

[8] Borrelli, F., Baotić, M., Bemporad, A. & Morari, M., 2005. Dynamic programming for constrained optimal control of discrete-time linear hybrid systems.. *Automatica*, Volume 41, pp. 1709-1721.

[9] Kwon, W. & Han, S., 2005. *Receding Horizon Control Model Predictive Control for State Models..* Berlin: Springer.

[10] Maciejowski, J. M., 2002. *Predictive control: with constraints.* s.l.:Pearson Education.

[11] Muske, K. R. & Rawlings, J. B., 1993. Model predictive control with linear models. *AICHE Journal*, 39(2), p. 262–287.

[12] Wang, L., 2009. *Model Predictive Control System Design and Implementation Using MATLAB®.* s.l.:Springer Verlag.

[13] Zhang, R., Xue, A. & Gao, F., 2019. *Model Predictice Control: Approached based on the extended state space model and extended non-minimal state space model.* s.l.:Springer.

[14] Kim, Y. & Bang, H., 2018. Introduction to Kalman Filter and Its Applications. . In: F. Govaers, ed. *In Introduction and Implementation of the Kalman Filter*. s.l.: Fraunhofer Institute for Communication, Informatio.

[15] State Estimation with kalman filter. Available at: tttp://techteach.no/fag/seky3322/0708/kalmanfilter/kalmanfilter.pdf