Journal of Engineering Technology

Volume 7, Special Issue (Internet of Things),Oct. 2018, PP. 273-291

RESEARCH ARTICLE -ENGINEERNG TECHNOLOGY

Computational Efficiency-based Comparison between Conventional MPC and Laguerre based MPC

Sohaib Aslam[1*], Salman Khalid[1], Mujtaba Jaffery[2], Sundas Hannan[2], Yasir Saleem[3], Sajid Sarwar[2]

[1]Department of Electrical Engineering, The Superior College (University campus) Lahore
[2]Department of Electrical Engineering, COMSATS University of Islamabad, Lahore Campus
[3]Department of Computer Science Engineering, University of Engineering and Technology, Lahore

Abstract: Introduction: Model Predictive Control (MPC) is an optimization-based control technique and handles process operational constraints effectively. However, high computational burden associated with optimization at every sampling instant is a constraint for its real-time embedded implementation. Therefore, recently researchers have worked on improving the computational efficiency of MPC. This paper presents a comparison between Conventional MPC and Laguerre functions based MPC in terms of computational efficiency with desired performance for both fast and slow dynamics systems i-e speed control of DC motor and temperature control of water tank respectively.

Methodology: The simulation results are done in MATLAB for the both cases by first developing the system model of DC motor and water tank and then they are controlled with desired performances by the Conventional and Laguerre based MPC techniques and their computational comparison is presented.

Results & Conclusion: The results have been tabulated in two cases. In first case speed control of DC motor is regulated by using both MPC techniques and computational comparison is done. In second case temperature control of water tank is regulated by MPC techniques and again computational comparison is done. The results have shown that the Laguerre based MPC provides better feasibility for real-time implementation of MPC for fast dynamic systems as theses systems require long control horizon to minimize steady-state error and Conventional MPC gives better feasibility to slow dynamic systems

Keywords: Computational burden, Control Horizon, Laguerre Functions, Laguerre Pole, Model Predictive Control, System Modeling

1 Introduction

Automatic Control System is of paramount importance in all fields of commercial ventures like quality control of made items, automatic mechanical production system, machine instruments, control space engineering, transportation frameworks, power frameworks, robotics and numerous more. The advancements in control technology have provided a platform to use more effective control techniques. Proportional Integral Derivative controller (PID) has been employed in more than 90 % of installed feedback systems [1]. Although mostly control processes have constraints, but PID cannot handle the process constraints effectively. Moreover, PID gains have to be adjusted repeatedly to get fine performance in dynamic environment [2]. Solution to these issues is Model Predictive Control (MPC). MPC reflects human behavior by selecting control activities which may lead to best anticipated result over some future range and regularly update choices as new perceptions are accessible [3]. MPC performs optimization at every sampling instant to obtain the control vector and only first element of control vector is applied to plant following a sliding time horizon strategy and its performance is dictated by two key parameters prediction horizon (N_p) and control horizon (N_c) [4,5]. MPC has become a popular control strategy for slow dynamic systems i-e petrochemical, servomechanism, crude atmospheric distillation and fluid catalic cracking etc because of high computational burden in numerical optimization and it becomes even more complicated for systems with fast dynamics and becomes non-feasible for real-time implementation on embedded systems with limited computational power [6,7]. However, recently research work has been done in developing an efficient MPC algorithm to make it feasible for real-time implementation [8]. Explicit solution of MPC to reduce computational burden has been in lime light for a decade and significantly increased the range of applications for MPC controllers [9]. Explicit solution does not replace optimization techniques but rather increase the application region of MPC by solving all optimization problems offline [10,11] and provide feasible solution for applications with sampling rates in μ-sec range [12,13]. The critical problem associated with explicit MPC is it does not reduce the computational burden of MPC technique so data storage may increase exponentially with increase in control horizon and prediction horizon and it becomes unsuitable for large dimensional problems [13]. In [14] the authors use an efficient search tree, but the offline computation of this can be prohibitive for complex controller partitions and storage requirements may even more increase. Efficient online MPC optimization is suitable for all areas of applications with low sampling rates to high sampling rates and from real-time PLC, FPGA implementations to simpler cost-effective microcontroller implementations. Therefore, Laguerre functions provide an opportunity to model the optimized control vector with less parameters in comparison to Conventional MPC (CMPC) and reduces online computational burden with desired closed loop system performance [15]. The aim of this paper is to lay down a comparison in terms of computational efficiency between CMPC and Laguerre based MPC (LMPC) for two cases i-e speed control of DC motor and temperature control of water tank. The results are developed in MATLAB by initially doing system modeling of DC motor and water tank and then implementing both MPC techniques and finally computational comparison is presented to estimate the improvement in computational efficiency by using Laguerre functions. The paper is organized as follows: section 2 describes the Laguerre functions and demonstrates its major application of system identification. In section 3 implementation of CMPC and LMPC is elaborated. Simulations based computational analysis is presented in section 4 and at the last conclusion of research work is summarized in section 5.

2 Laguerre Functions

The optimized control vector ΔU of MPC is represented by Eq. (1)

$$\Delta U = [\Delta u(k_i)\ \Delta u(k_i+1) \ldots\ldots \Delta u(k_i+N_C-1)]^T \tag{1}$$

Where, N_C is the control horizon and it is the number of elements in control vector. At time k_i, the elements of vector can be described as discrete δ-functions with ΔU is the coefficient as shown in Eq. (2)

$$\Delta u(k_i+i) = [\delta(i)\ \delta(i-1) \ldots\ldots \delta(i-N_c+1)]\Delta U \tag{2}$$

Where, δ(i) = 1 if i = 0 and δ(i) = 0 if i ≠ 0 so the function δ is the pulse operator and δ(i-d) shifts the center of pulse onward with increase in index d so above relation can be used to capture the trajectory of control vector. It is apparent that $\Delta u(k_i+i)$ for i=0, 1,….N_C-1 can be expressed by discrete polynomial function so discrete Laguerre functions can be used to realize the relation shown in Eq. (1) [16].

2.1 Discrete-Time Laguerre Networks

The Laguerre functions are orthonormal and discrete time Laguerre network is developed by discretization of a continuous-time Laguerre network. The set of continuous time Laguerre functions for any value of p > 0 are illustrated below [16]:

$$l_1(t) = \sqrt{2p} \times e^{-pt}$$

$$l_2(t) = \sqrt{2p}(-2pt+1)e^{-pt}$$

$$l_i(t) = \sqrt{2p}\frac{e^{pt}}{(i-1)!}\frac{d^{i-1}}{dt^{i-1}}[t^{i-1}e^{-2pt}] \quad (3)$$

Where, p is the time scaling constant of Laguerre functions. The z-transforms of discrete-time Laguerre functions are shown in Eq. (4)

$$r_1(z) = \frac{\sqrt{1-a^2}}{1-az^{-1}}$$

$$r_2(z) = \frac{\sqrt{1-a^2}}{1-az^{-1}}\frac{z^{-1}-a}{1-az^{-1}}$$

$$r_N(z) = \frac{\sqrt{1-a^2}}{1-az^{-1}}(\frac{z^{-1}-a}{1-az^{-1}})^{N-1} \quad (4)$$

Where, a is the user define Laguerre pole and its value ranges from 0 to 1. The discrete time Laguerre functions can be easily realized using state space approximation of network as given in Eq. (5)

$$r_k(z) = r_{k-1}(z)\frac{z^{-1}-a}{1-az^{-1}} \quad (5)$$

Let denote the inverse z-transform of $\Gamma_1(z,a)$ by $l_1(k)$, $\Gamma_2(z,a)$ by $l_2(k)$ and similarly $\Gamma_N(z, a)$ by $l_N(k)$ and this forms a set of Laguerre functions in discrete domain shown in a vector form as shown in Eq. (6)

$$L(k) = [l_1(k) l_2(k) \ldots \ldots l_N(k)]^T \quad (6)$$

Taking advantage of Eq. (4) the set of discrete time Laguerre functions validates the following difference Eq. (7) [17,18]

$$L(k+1) = A_l L(k) \quad (7)$$

Where, A_l is (N×N) matrix and it is a function of parameters a and $\beta = (1-a^2)$. By analysing the Laguerre equations and using the following z transform properties given in Table 1

Table 1. Properties of Z Transform

Sr. No.	Z-Domain	Discrete-time
1	X(z)	X(k)

2	X(z)H(z)	$\sum_{m=0}^{\infty} x(m)h(k-m)$
3	$\lim_{z\to 1} X(z)$	X(0)

One can obtain the initial condition as shown in Eq. (8) [16].

$$L(0)^T = \sqrt{\beta}[1 - a \;\; a^2 - a^3 \ldots \; (-1)^{N-1}(a)^{N-1}] \qquad (8)$$

For N=5 L(0) and A_l will become

$$A_l = \begin{bmatrix} a & 0 & 0 & 0 & 0 \\ \beta & a & 0 & 0 & 0 \\ -a\beta & \beta & a & 0 & 0 \\ a^2\beta & -a\beta & \beta & a & 0 \\ -a^3\beta & a^2\beta & -a\beta & \beta & a \end{bmatrix}; L(0) = \sqrt{\beta}\begin{bmatrix} 1 \\ -a \\ a^2 \\ -a^3 \\ a^4 \end{bmatrix}$$

2.2 Application of Laguerre Functions in System Modeling

One of the significant areas of discrete-time Laguerre functions is system identification, where impulse response of a system is modeled by Laguerre functions. Consider the impulse behavior of a stable system H(k) shown in Eq. (9) [16].

$$H(k) = C_1 l_1(k) + C_2 l_2(k) + \cdots + C_N l_N(k) \qquad (9)$$

Where, the coefficients C_1, C_2, C_3,….C_N are calculated from the system data and N is the number of parameters used. These coefficients can be determined by the following relation.

$$C_i = \sum_{k=0}^{\infty} H(k) l_i(k) \qquad (10)$$

Let us consider a system with z–transfer function given below:

$$G(z) = \frac{z - 0.1}{[(z - 0.8)(z - 0.9)]}$$

Model the impulse response of this system and monitor the effect of changing N and a on the identification of the system behavior. At first generate the initial conditions of Laguerre functions by putting the value of a=0.8 and N=3 in Eq. (8)

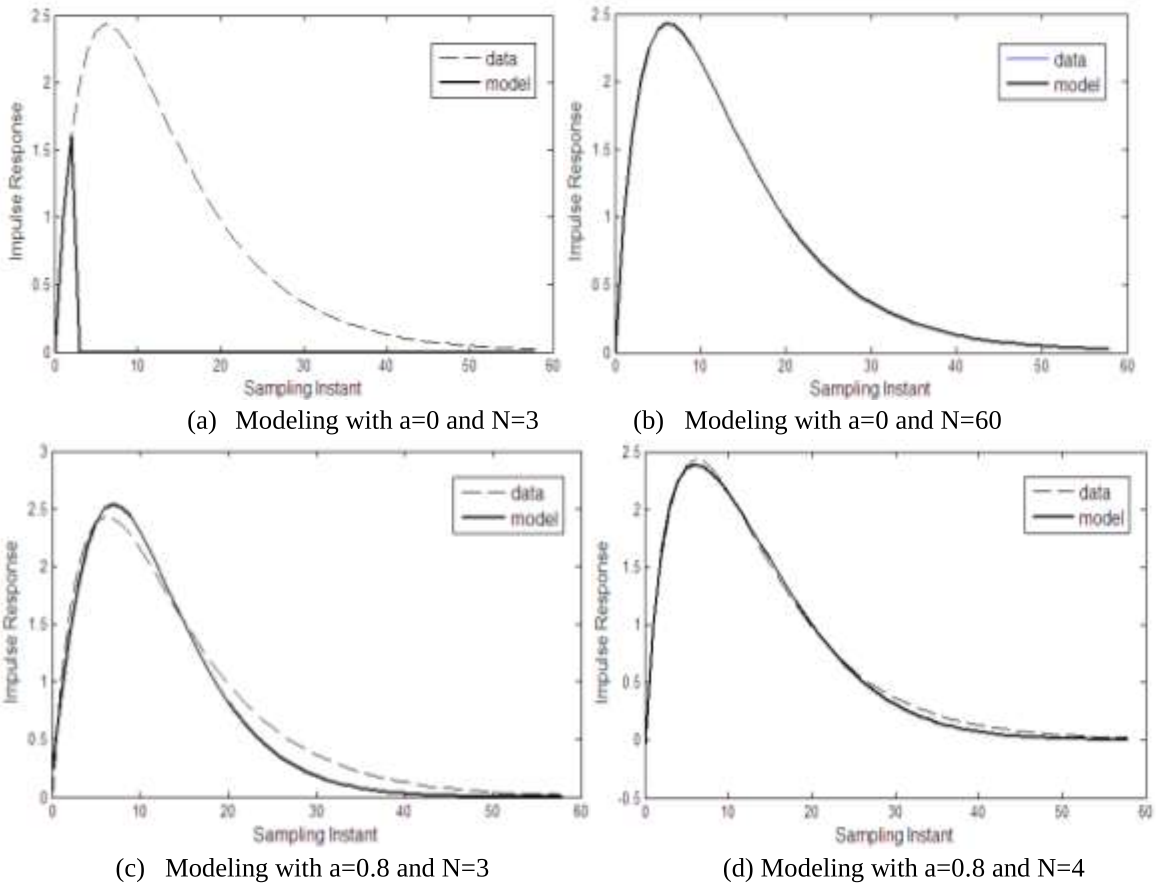
$$L(0) = [0.60 \; -0.48 \;\; 0.38]^T$$

$$A_l = \begin{bmatrix} 0.8 & 0 & 0 \\ 0.3 & 0.8 & 0 \\ -0.2 & 0.3 & 0.8 \end{bmatrix}$$

Laguerre coefficients are calculated by using Eq. (10)

$$\eta = [C_1 C_2 C_3] = [4.38\ 6.86\ 2.37]$$

By putting the values of Laguerre functions and Laguerre coefficients in Eq. (9) the system model is identified. The results are shown in Figure 1



(a) Modeling with a=0 and N=3 (b) Modeling with a=0 and N=60

(c) Modeling with a=0.8 and N=3 (d) Modeling with a=0.8 and N=4

Figure. 1. Comparison between Actual Data and Laguerre Model

It is observed from Figure 1 that when a=0 and N=3 the approximated model does not represents the actual data so to model the actual data with a=0 large number of parameters are required (N=60) which results in high computational burden. To decrease the computational burden both a and N are used in system identification. The above figure also shows that with a=0.8 and N=4 improved results have been achieved and only 4 parameters are required to approximate the system model. The optimization of ΔU control vector in MPC relates to pulse operator with parameter a = 0 in the Laguerre polynomial. Consequently, control systems with high sampling rate and high-performance requirements will require a large number of parameters for optimization of the control signal ΔU and results in high computational burden making MPC less feasible for real-time implementation on simple embedded systems. Instead, a more practical approach is to employ Laguerre networks in the design of MPC.

3 Implementation of MPC

Implementation MPC refers to a class of control algorithms that utilizes process model to predict the future response of a plant. At each sampling instant MPC optimizes future plant behavior by computing a sequence of controlled inputs. The first element in the optimal control sequence is applied to the plant while rest of sequence is ignored, and this entire calculation is repeated at every sampling instant [17]. The basic structure of MPC is shown in Figure 2.

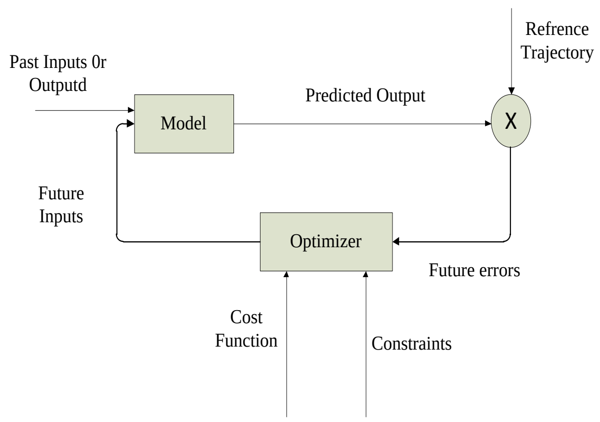


Figure. 2. Basic structure of MPC

Implementation of MPC is based on system model and optimizer. Optimizer is a mathematical function which optimizes the control signal by minimizing the cost function in presence of constraints. These control signals as future inputs are send to system model of the plant to get predicted output. The difference between the predicted output and reference value in form of errors is sent to optimizer which optimizes the future input and this procedure will remain continue.

3.1 System Modeling

Perfection of MPC depends on how well dynamics of a system have been captured. In this research, computational burden of CMPC and LMPC is evaluated for the case of speed control of DC motor and temperature control of water tank. The choice of these two systems is based on the dynamics of these two systems. Speed control of DC motor is considerably fast system as compared to temperature control of water tank. Therefore, provides an opportunity to understand the feasibility for real-time implementation of MPC for both slow and fast systems. The system modeling for theses two systems is elaborated below:

Journal of Engineering Technology

Volume 7, Special Issue (Internet of Things),Oct. 2018, PP. 273-291

RESEARCH ARTICLE -ENGINEERNG TECHNOLOGY ASEE

3.1.1 System Model for Speed Control of DC motor

There are four types of DC motor i-e shunt, series, permanent magnet and compound. Permanent magnet DC motor (PMDC) is used in this research work. DC motor model is shown in Figure 3. The input voltage, armature resistance, armature inductance, back emf produced by motor, moment of inertia, motor torque, speed of motor and load torque are represented by V_m, R_m, L_m, V_b, J_m, T_m, ω_m and T_L respectively [19].

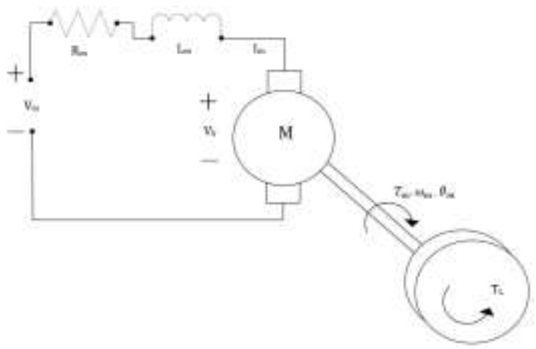


Figure. 3. DC motor

Apply KVL to above figure as shown in Eq. (11)

$$V_m = R_m I_m + L_m \frac{di(t)}{dt} + k_m \omega_m \tag{11}$$

Where V_b = k_ω and k_m= Motor torque constant. Motor is an electromechanical system and it can be represented by equation of motion as shown in the Eq. (12).

$$\frac{Jd\omega_m(t)}{t} = T_m - T_L - B_m \omega_m \tag{12}$$

Here T_L is assumed zero to simplify transfer function modeling, B_m is damping friction and $T_m = k_m I_m$
After applying Laplace transform to Eq. (11) and (12), simultaneously solve them to get the transfer function model as shown in Eq. (13).

$$\frac{\omega_m}{V_m} = \frac{k_m}{[(L_m J_m)S^2 + (R_m J_m + B_m L_m)S + (k_m^2 + R_m B_m)]} \tag{13}$$

Substituting DC motor parameters from Table 2 in Eq. 13, the transfer function becomes [19]

$$\frac{19.607}{[(0.00014)S^2+(0.0049)S+1]} \tag{14}$$

Table 2. Internal Parameters of DC Motor [19]

Sr. No.	Parameter	Value
1	Moment of inertia	J_m = 0.00025 N_m/rad/s^2
2	Damping factor	B_m = 0.0001 N_m/rad/s
3	Armature resistance	R_m = 0.5 Ω
4	Armature inductance	L_m = 1.5 mH
5	Torque constant	K_m = 0.05 N_m/A
6	Rated voltage	V_m = 12 V
7	Rated speed	2400 RPM

This transfer function is discretized by some sampling interval. The sampling interval of a control system is calculated by using a rule of thumb given below [1]:

$$\frac{1}{T} > 30 \times BW\ of\ System$$

Where BW is the bandwidth of system and it is directly calculated by MATLAB to be 124 Hz. By putting the value of BW in above relation sampling time is 268 µs. The State Space Model (SSM) of DC motor for speed control is shown in Eqs. (17) and (18)

$$x_m(k+1) = \begin{bmatrix} 1.9908 & -0.9912 & 0.0045 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} x_m(k) + \begin{bmatrix} 0.0045 \\ 0 \\ 1 \end{bmatrix} u(k) \tag{17}$$

$$y(k) = [1 \quad 0 \quad 0] x_m \tag{18}$$

3.1.2 System Model for Temperature Control of Water Tank

The quality of heating process is based on induction, maintenance and control. There are number of heating techniques in practice now days for domestic and industrial processes i.e. direct heating, indirect heating, electric arc heating etc. In this research work direct resistance heating method is applied where immersion rod is dipped in the water tank. The heating capacity of immersion rod has a direct relation with its wattage power and the required wattage power for the heating rod is calculated by the following equation [20].

$$Q_T = \frac{WC\Delta T_{water}}{3412.T} \tag{19}$$

Journal of Engineering Technology

Volume 7, Special Issue (Internet of Things),Oct. 2018, PP. 273-291

RESEARCH ARTICLE -ENGINEERNG TECHNOLOGY

Where QT, W, C, ΔT_{water} and T are heater wattage, weight of water, specific heat of water, change in temperature and time required to raise temperature at desired point respectively. In this research work W is *33* lb, C_P is 1 Btu/lb/F, ΔT is 140 F, and T is 1 hour. Substitute these values in Eq. (19) to calculate required wattage of immersion rod

$$Q_T = \frac{88 * 1 * 140}{3412 * 1} = 1.8\ KW$$

Required power of immersion rod is 1.8KW so 2KW heating rod is used. This heating rod is used to control the temperature of water tank system by using MPC. Performance of MPC depends on accuracy of system modelling. The system model of water tank is shown in Figure 4

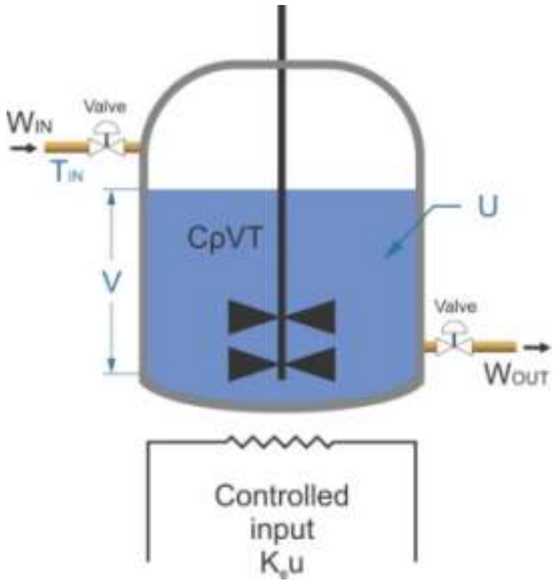


Figure. 4. Water Tank Model

Where $K_e u$ is the input power, ρ is the density of water, $C\rho VT_1$ is the energy in the tank, V is the volume, T1 is the water temperature in the tank, T_{in} is the inlet temperature, W is the mass flow rate of water and U is the thermal conductivity. The parameter values of liquid tank are shown in Table 3 [21].

Table3. Parameters used in simulations

Sr. No.	Parameter	Value
1	Weight	K_{tau} = 40Kg
2	Inlet Flow rate	W_{in} = 20Kg/min
3	Input Power	K_e = 2KW
4	Specific Heat	C = 4200J/Kg-K
5	Thermal	U = 0.6W/mK @
6	Density	ρ = 1000Kg/m^3
7	Volume	V = 0.20m^3

The thermodynamic equation for water tank is shown in Eq. (20) [21]

$$\frac{d(C\rho VT_1)}{dt} = K_e u + cW(T_{in} - T_1) + U(T_{env} - T_1) \tag{20}$$

By taking Laplace of above equation it becomes

$$C\rho V[sT_1(s) - T_{1o}] = K_e u(s) + CW\big(T_{in}(s) - T_1(s)\big) + U\big(T_{env}(s) - T_1(s)\big) \tag{21}$$

Assumed that $T_{in} = T_{env}$

$$C\rho V[sT_1(s) - T_{1o}] = K_e u(s) + (Cw + U)\big(T_{in}(s) - T_1(s)\big) \tag{22}$$

Solving for $T_1(s)$ above equation becomes

$$T_1(s) = \frac{{K_e}/{CW+U}}{\frac{\rho V}{(w+{U}/{c})}s+1}u(s) + \frac{1}{\frac{\rho V}{(w+{U}/{c})}s+1}T_{in}(s) \ + \frac{{\rho V}/{(W+{U}/{C})}}{\frac{\rho V}{(w+{U}/{c})}s+1}T_{1o} \tag{23}$$

τ is the time delay between an excitation of heating element and the response of temperature sensor.

$$T(t) = T_1(t - \tau) \tag{24}$$

Apply Laplace transform to Eq. (24) and it becomes

$$T(s) = T_1(s)e^{-\frac{K_\tau}{W}s} \tag{25}$$

By putting $T_1(s)$ in above equation, the transfer function is shown in Eq. (26)

$$\frac{T(s)}{u(s)} = \frac{{K_e}/{cw+U}}{\frac{\rho V}{(w+{U}/{c})}s+1}e^{-\frac{K_\tau}{w}s} \tag{26}$$

Here $\tau = \frac{K_\tau}{w}$, $K_u = \frac{K_e}{cw+U}$, $T_k = \frac{\rho V}{(w+{U}/{c})}$ so transfer function H(s) in standard first order is shown in Eq. (27)

$$H(s) = \frac{K_u}{T_k s+1}e^{-\tau s} \tag{27}$$

By putting the values of parameters from Table 3 to above equation, transfer function becomes

$$H(s) = \frac{0.02}{10s+1}e^{0.03s} \tag{28}$$

The transfer function of the water tank for temperature control is also discretized and its sampling time is calculated by the above relations is 10sec representing a system with slow dynamics. The SSM of water tank for temperature control is represented by Eqs. (29) and (30)

$$x_m(k) = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} x_m(k) + \begin{bmatrix} 0.0200 \\ 0 \\ 0 \end{bmatrix} u(k) \tag{29}$$

$$y(k) = [1 \quad 0 \quad 0]x_m(k) \tag{30}$$

3.2 Optimization

The optimizer in MPC structure is a mathematical function which minimizes the cost function by reducing the difference between set-point and real value and in return optimizes control vector ΔU. The performance of MPC is based on two performance parameters N_P and N_C. particularly important for systems with fast dynamics but their values are carefully chosen as they have direct relation with computational burden associated with MPC. Therefore, Laguerre functions are used to model the control vector by replacing the use of Nc with Laguerre pole a and performance parameter N. The optimization techniques for both CMPC and LMPC are elaborated below:

3.2.1 Conventional MPC Optimization

MPC optimizes the control vector ΔU by minimizing the cost function with predetermined constraints as shown in Eq. (31) [21]

$$J = (R_s - Y)^T(R_s - Y) + \Delta U^T \bar{R} \Delta U \tag{31}$$

The first term in Eq. (31) reduces the difference between the set-point and the predicted output while, second term pays attention to change in control signal ΔU. $\overline{\mathrm{R}}$ is a diagonal matrix and $\overline{\mathrm{R}} = \mathrm{r_w} \mathrm{I}_{\mathrm{N_c}\times\mathrm{N_c}}$. $\mathrm{r_w} \geq 0$ is a user defined tuning parameter for desired closed loop performance. When $r_w = 0$ is in cost function no importance is given to change in control signal and large value of r_w produces careful change in control signal. The optimized control vector is shown in Eq. (32) [21].

$$\Delta U = (\phi^T\phi + \bar{R})^{-1}\phi^T\big(R_s F x(k_i)\big) \tag{32}$$

Where F and ϕ are the vectors shown below

$$F = \begin{bmatrix} CA \\ CA^2 \\ \cdot \\ \cdot \\ CA^{N_p} \end{bmatrix} \quad \text{and } \phi = \begin{bmatrix} CB & 0 & 0 & 0 & \dots & 0 \\ CAB & CB & 0 & 0 & \dots & 0 \\ \vdots & CAB & CB & 0 & \dots & 0 \\ CA^{N_p-1}B & CA^{N_p-2}B & CA^{N_p-3}B & \dots & \dots & CA^{N_p-N_c}B \end{bmatrix}$$

There are three forms of constraints two of them are applied at the control signal and rate of change in control signal while, the third one is applied at the output of the system. These constraints are presented in Eq. (33) [21]

$$\Delta U^{min} \leq \Delta U(k) \leq \Delta U^{max} \tag{33a}$$

$$u^{min} \leq u(k) \leq u^{max} \tag{33b}$$

$$y^{min} \leq y(k) \leq y^{max} \tag{33c}$$

Normally the constraints are applied on the whole control vector and all constraints are expressed in terms of vectorΔU as shown in Eq. (34)

$$\begin{bmatrix} u(k_i) \\ u(k_i+1) \\ u(k_i+2) \\ \vdots \\ u(k_i+N_c-1) \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} u(k_i-1) + \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ 1 & 1 & 0 & \dots & 0 \\ 1 & 1 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \dots & 0 \\ 1 & 1 & \dots & 1 & 1 \end{bmatrix} \begin{bmatrix} \Delta u(k_i) \\ \Delta u(k_i)+1 \\ \Delta u(k_i)+2 \\ \vdots \\ \Delta u(k_i)+N_c-1 \end{bmatrix} \tag{34}$$

Rearranging the above equation in a compact matrix form by relating C1 and C2 to appropriate matrices and subject to constraints for control signal as shown in Eq. (35)

$$-(C_1 u(k_i-1) + C_2\Delta U) \leq -U^{min} \tag{35a}$$

$$(C_1 u(k_i-1) + C_2\Delta U) \leq U^{max} \tag{35b}$$

Where U^{min} and $U^{max}$are column vectors with Nc elements of u_{min} and u_{max} . Similarly for change in control signal constraints are shown in Eq. (36)

$$-\Delta U \leq -\Delta U^{min} \tag{36a}$$

$$\Delta U \leq \Delta U^{max} \tag{36b}$$

Again ΔU^{min} and ΔU^{max} are column vectors with N_c elements. Similarly, output constraints are presented in terms of ΔU as shown in Eq. (37)

$$Y^{min} \leq Fx(k_i) + \phi\Delta U \leq Y^{max} \tag{37}$$

After minimizing the cost function for optimization of control vector, ΔU is subjected to the inequality constraints by Eq. (38)

$$\begin{bmatrix} M_1 \\ M_2 \\ M_3 \end{bmatrix} \Delta U \leq \begin{bmatrix} N_1 \\ N_2 \\ N_3 \end{bmatrix} \tag{38}$$

Where, $M_1 = \begin{bmatrix} -C_2 \\ C_2 \end{bmatrix}$; $N_1 = \begin{bmatrix} -U^{min} + C_1 u(k_i-1) \\ U^{max} - C_1 u(k_i-1) \end{bmatrix}$; $M_2 = \begin{bmatrix} -1 \\ 1 \end{bmatrix}$; $N_2 = \begin{bmatrix} -\Delta U^{min} \\ \Delta U^{max} \end{bmatrix}$;

$M_3 = \begin{bmatrix} \phi \\ \phi \end{bmatrix}$; $N_3 = \begin{bmatrix} -Y^{min} + Fx(k_i) \\ Y^{max} - Fx(k_i) \end{bmatrix}$

Transform the above equation in more compact form as shown in Eq. (39) below and use it to find an optimized control vector without violating the constraints [21].

$$M\Delta U \leq \gamma \tag{39}$$

Where, M is a vector of constraints with its rows equal to number of constraints and columns equal to dimension of ΔU.

3.2.2 Laguerre Based MPC Optimization

In LMPC, the Eq. (32) is interpreted in terms of Laguerre functions, as constraints are not considered in this research work optimization relation becomes simple as shown in Eq. (40) [1]

$$\Delta U(k_i) = L_i(k)^T \eta_i \tag{40}$$

Where $L_i(k)$ is Laguerre function and η is a function calculated at every sampling instant to optimize the control vector as shown in Eq. (41) [1]

$$\eta = -\left(\sum_{m=1}^{N_p} \phi(m) Q \phi(m)^T + R_L\right)^{-1} . \left(\sum_{m=1}^{N_p} \phi(m) Q A^m\right) x(k_i) \tag{41}$$

Q and R are weight matrices, A is system matrix and $\phi(m)$is a function expressed in Eq. (42)

$$\phi(m) = \sum_{i=0}^{m-1} A^{m-i-1} B_L(i) \tag{42}$$

In above relation B is the input matrix of system and it is clear from above relations that N_C has no effect on system performance which results an improvement in computational efficiency of LMPC.

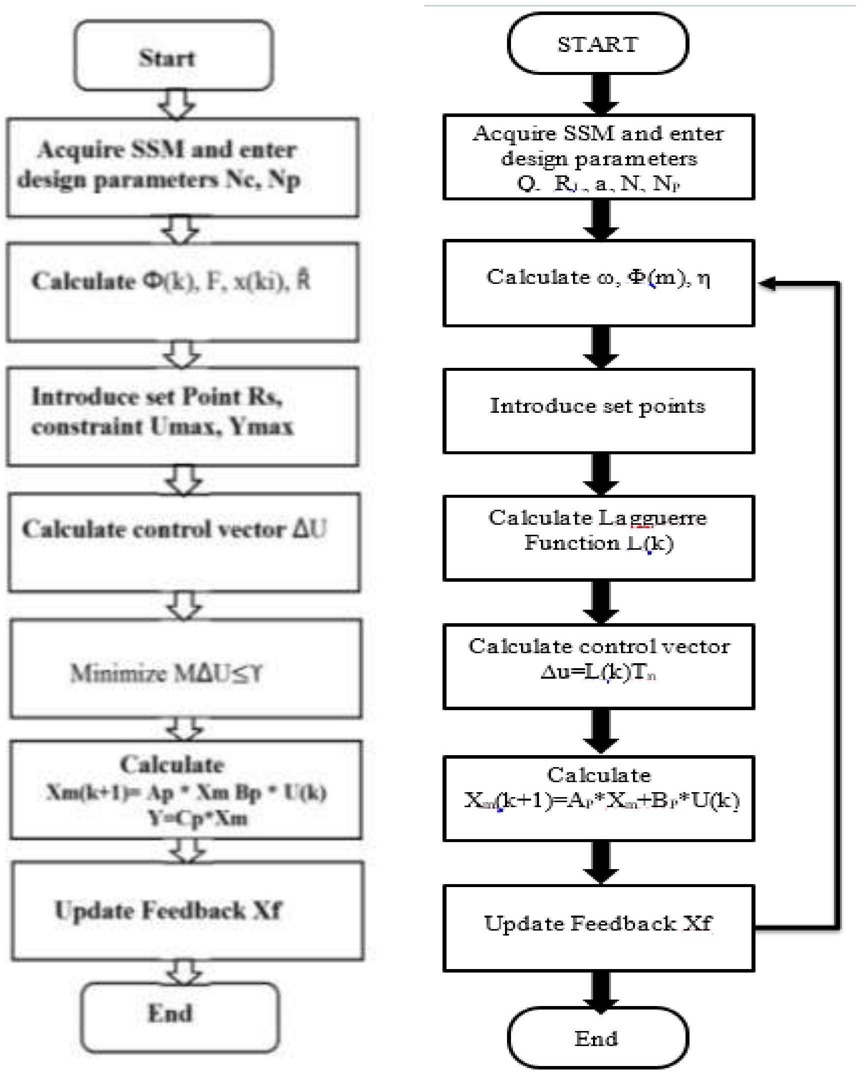
4 Simulations based Performance Analysis

Simulations based computational analysis between CMPC and LMPC is presented for two systems one has fast dynamics i-e speed control of DC motor and other has slow dynamics i-e temperature control of water tank. The computational comparison is done for two calculations one is initial calculation of variables used in optimization equations and other is calculation of optimized control vectorΔU.

4.1 MPC Implementation Flowchart

The Simulations of CMPC and LMPC are developed by first developing the state-space model (SSM) of both DC motor and water tank then include the design parameters Nc, Np, Laguerre pole a, performance parameter N, weights Q and R and reference point. Finally, the cost function is minimized to optimize control vector which is further applied to SSM to get predicted output and this procedure continues until the desired set point is achieved. Flow chart of MPC implementations are shown in Figure 5

Volume 7, Special Issue (Internet of Things),Oct. 2018, PP. 273-291



(a) CMPC implementation hierarchy (b) LMPC implementation hierarchy

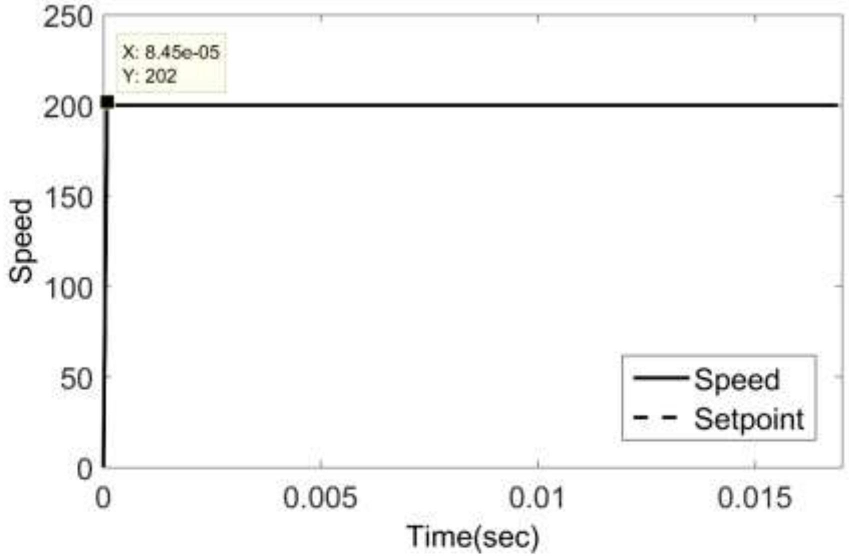
Figure. 5. MPC implementation flow charts

4.2 Simulation Results

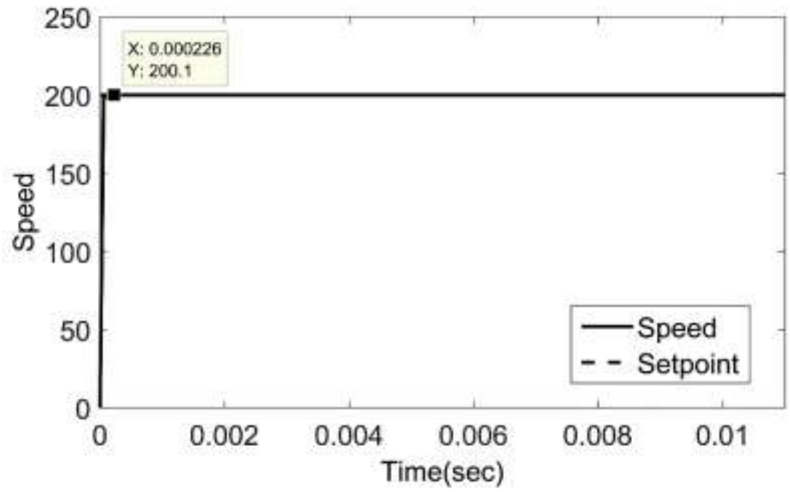
The simulation results are presented for two cases given below:

4.2.1 Speed Control of DC motor

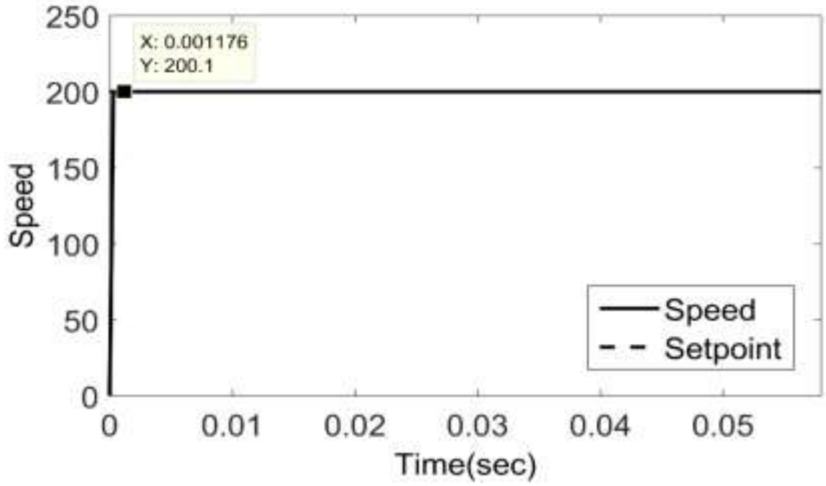
The Speed of DC motor is regulated at 200 rad/sec and the execution time required to calculate initial variables and optimization equations for both techniques is calculated by using tic-toc command in MATLAB. To avoid error in estimating computational time as there is no dedicated processor for MATLAB average computational time is measured with 1000 loops. The results are shown in Figure 6 and Figure 7



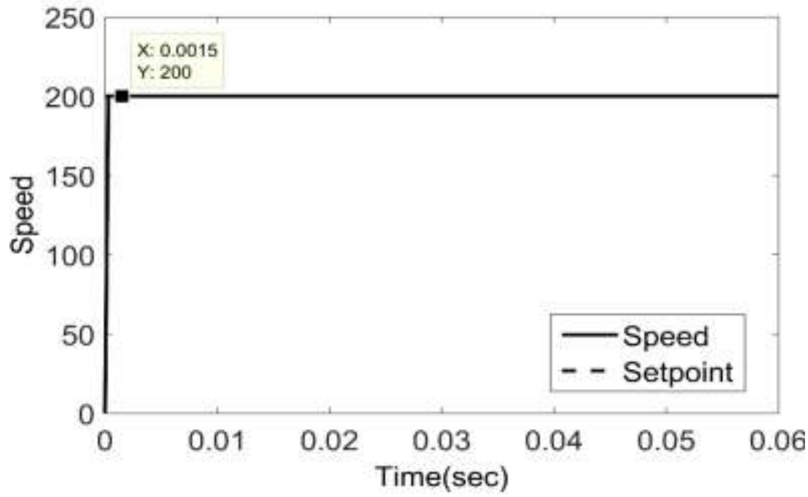
(a) CMPC based speed control with N_P =10 and N_C=2



(b) CMPC based speed control with N_P =10 and N_C=5



(c) CMPC based speed control with N_P =15 and N_C=15



(d) CMPC based speed control with N_P =20 and N_C=20

Figure. 6. CMPC based speed control with different Np and Nc

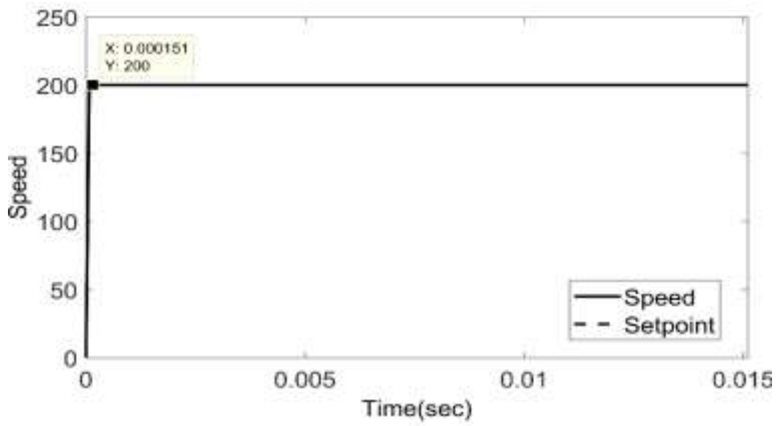


Figure. 7. LMPC based speed control with a=0.5 and N=1

From Figure 6 it is clear that speed regulation of DC motor at 200rad/sec without overshoot requires N_C=20 and N_P=20 and these two variables have direct relation with computational burden specifically N_C. Therefore, increase in N_C increases computational time for both initial calculations and optimization calculations. Fig. 7 shows speed control of DC motor by LMPC. The computational comparison between these two techniques is presented in Table 4

Table4. Computational comparison between CMPC and LMPC for speed control of DC motor

Case	Horizon	Horizon	Pole	Parameter (a) (N)	Overshoot (OS)	Computational Time (µs)	
						CT1	CT2
CMPC	2	10	N/A	N/A	2	75µs	29µs
	5	10	N/A	N/A	0.1	149µs	33µs
	15	15	N/A	N/A	0.1	344µs	98µs
	20	20	N/A	N/A	0	564µs	111µs
LMPC	N/A	N/A	0.5	1	0	98µs	44µs

In Table 4 CT1 shows computational time to calculate initial variables and CT2 shows optimization computational time so it is evident from above table that N_C and N_P have to be carefully chosen to optimize the control vector as they increase the computational burden. However, LMPC effectively improves both CTs and provides a feasible solution for real-time implementation. It is also clear from above table that for 0% overshoot there is a huge difference in CT for both techniques thus, convincing the superiority of LMPC over CMPC for systems having fast dynamics

4.2.2 Temperature Control of Water tank

As Temperature of water tank is regulated at desired temperature 200 °C. The results for both techniques are shown in Fig. 8 and Fig. 9

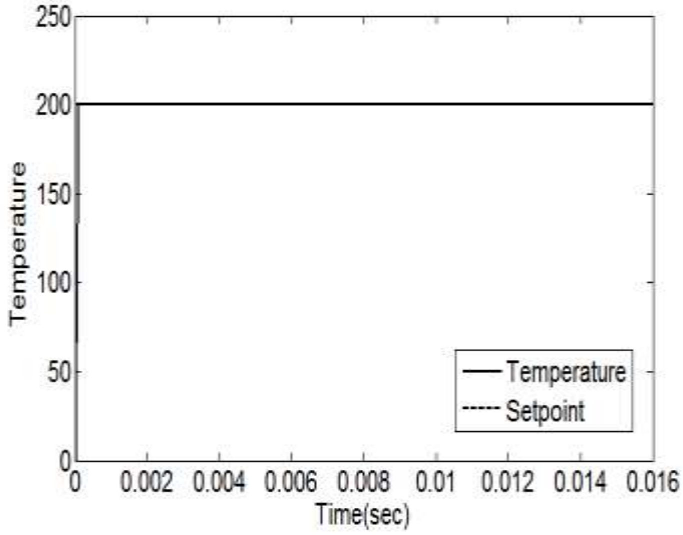


Figure. 8. CMPC based temperature control with Np=10 and Nc=2

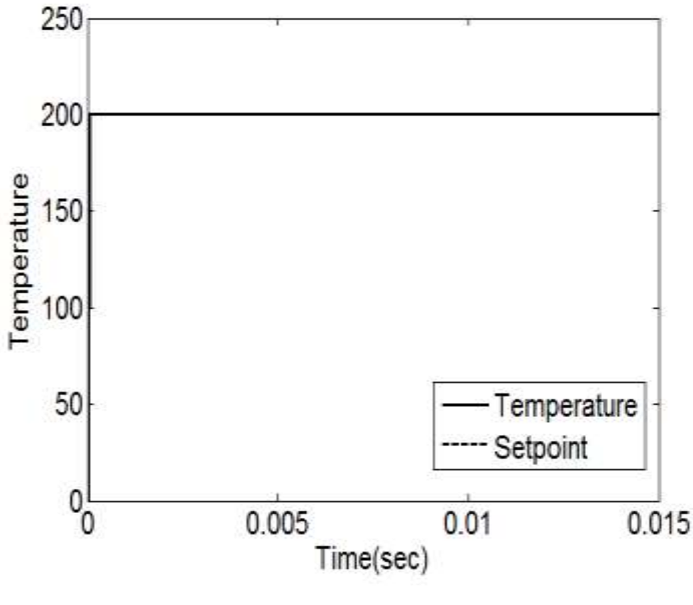


Figure. 9. LMPC based temperature control with a=0.5 and N=1

From above figures, it is clear that both techniques effectively regulate the temperature without any overshoot and the Fig. 8 also gives an idea that CMPC can regulate a slow dynamic system with small value of N_C and N_P which results in less computational burden and consequently CMPC becomes feasible for real-time implementation. The computational comparison between CMPC and LMPC for temperature control of water tank is summarized in Table 5

Table5. Computational comparison between CMPC and LMPC for temperature control of water tank

Case	Horizon (N	Horizon (N	Pole (a)	Parameter (N)	Overshoot (OS)	Computational Time (µs)	
CMPC						CT1	CT2
	2	10	N/A	N/A	0	62µs	26µs
LMPC	N/A	N/A	0.5	1	0	85µs	44µs

Table 5 convincingly supports the application of CMPC for slow dynamic systems as they can be effectively controlled with small values of N_C and N_P and makes it possible for real-time implementation on small embedded systems and even CMPC is more computational efficient than LMPC for the case temperature control of water tank.

5 Conclusion

In this research work computational efficiency-based comparison between Conventional MPC (CMPC) and Laguerre based MPC (LMPC) is done. Recent research shows that CMPC is best suited for systems with slow dynamics. Therefore, comparison is analyzed for both fast and slow systems i-e speed control of a DC motor and temperature control of water tank respectively. The results have shown that CMPC effectively handles slow systems with high sampling times and requires less computational time for optimization of control vector. Moreover, CMPC proves to be more computationally efficient than LMPC for temperature control of water tank. Therefore, it becomes very much feasible for real-time implementation. However, for case of speed control of a DC motor CMPC requires higher values of its performance parameters i-e N_C and N_P to regulate speed without overshoot which results in increased computational burden hence making it infeasible for real-time implementation on simple embedded systems. On the other hand, LMPC effectively reduces computational burden and declares itself an optimum solution for real-time implementation for fast dynamic systems. In future prospect, this comparison can be tested by implementing both techniques on a simpler 16-bit microcontroller. Moreover, LMPC can be compared with other latest computationally efficient optimization techniques for some MIMO systems and can be tested as a prototype by using MPLAB and Proteus software.

References

[1] S. Aslam, M. Jaffery, "Real-time implementation of model predictive control on a 16-bit microcontroller for speed control of a DC motor " Journal of Engineering Technology, vol. 6, pp. 415-434, 2017

[2] K. S. Holkar, L.M. Waghmere, K.K. Wagh, "An Overview of Model Predictive Control", International Journal of Control and Automation, vol. 3, pp.47-64, December, 2010

[3] J. Rossister, R.H. Bishop**.** Model based Predictive Control a Practical approach. CRC Press, 2005.

[4] P. Verlade, J. M. Maestre, C. Ocampo-Martinez, C. Bordons, "Application of Robust Model Predictive Control to a Renewable Hydrogen-based MicroGrid" IEEE Europian Control Conference, pp. 1209-1214, July, 2016

[5] A.K. Abes, F. Bouani, M. Kasouri, "A microcontroller Implementation of Constrained Model Predictive Control" World Academy of Science, Engineering and Technology, vol 5, pp. 662-665, 2011

[6] S. E. Li, Z. Jia, K. Li, B. Cheng, "Scale Reduction based Efficient Model Predictive Control and Its Application in Vehicle Following Robot" Proceedings of 16th IEEE International Annual conference on Intelligent Transportation Systems, pp. 1266-1271, October, 2013.

[7] S.E.Oltean, M.Dulau, A.V.Duka, "Model reference Adaptive Control Design for Slow Processes. A case study on level process control", Procedia Technology, vol. 22, pp. 629-636, 2015

[8] A. A. Kherji, F. Bouani, M. Ksouri, "A MICROCONTROLLER IMPLEMENTATION OF MODEL PREDICTIVE CONTROL" WORLD ACADEMY OF SCIENCE, ENGINEERING AND TECHNOLOGY, VOL. 56, PP. 662-665, 2011

[9] A. Bemporad, M. Morari, V. Dua, E. N. Pistikipoulos, "The Explicit linear quadratic regulator for constrained systems " Automatica, vol. 38, pp. 1845-1846, 2002.

[10] M. Canale, L. Fagiano, M. Milanese, "Set membership approximation theory for fast implementation of model predictive control laws" Automatica, vol. 43, pp. 1163-1168, 2009

[11] F. J. Christophersen, M. Kvasnika, C. N. Jomes, M. Morari, "Efficient evaluation of piecewise control laws over a large number of polyhedra" IEEE Conference on European Control, pp. 2360-2367, 2007

[12] G. Valencia-Palomo, J. A. Rossister, "Auto-tuned predictive control based on minimal plant information " International Symposium on Advanced Control of Chemical Processes, pp. 554-559, 2009

[13] G. Valencia-Palomo, J. A. Rossister, "Using Laguerre functions to improve efficiency of multi-parametric predictive control" IEEE American Control Conference, pp. 4731-4736, 2010.

[14] P. Tondel, T. A. Johansen**, A. Bemporad**, "Evaluation of Piecewise offline control via binary search tree" Automatica, vol. 39, pp. 945-950, 2003

[15] J.A. Rossister, L. Wang, "Exploiting Laguerre functions to improve the feasibility/performance compromise in MPC" IEEE Conference on Decision and Control, 2008

[16] L. Wang. Model Predictive Control system design and implementation using MATLAB. Springer, 2009

[17] M. Chipofya, D. J. Lee, K. T. Chong, "Trajectory Tracking and Stabilization of a Quadrotor using Model Predictive Control of Laguerre functions" Abstract and Applied Analysis, vol. 2015, pp. 1-11, 2015

[18] M.Spacic, D.Mitic, M.Hovd, D.Antic, "Tube Model Predictive Control based on Laguerre functions with an Auxialiary Sliding Mode Controller", IEEE 15th International Symposium on Intelligent Systems and Informatics, pp. 243-248, 2017.

[19] S. Aslam, S. Hannan, W. Zafar, U. Sajjad, “Implementation of PID on PIC24F series Microcontroller for speed control of a DC motor using MPLAB and Proteus” Advances in Science and Technology Research Journal, vol. 10, pp. 40-50, September, 2016

[20] O. P. Rajesh, K. rajesh, “Intelligent temperature controller for water-bath system ” International Journal of Computer, Information, Systems and Control Engineering, vol. 6, pp.1178-1184, 2012

[21] S.Aslam, S.Hannan, W.Zafar, U.Sajjad, “Temperature Control of Water-Bath System in Presence of constraints by using MPC” International Journal of Advanced and Applied Sciences, vol.3, pp62-68, 2016