IJCSN International Journal of Computer Science and Network, Volume 4, Issue 4, August 2015
ISSN (Online) : 2277-5420 www.IJCSN.org
Impact Factor: 0.417

A Comparison of Constrain Model Predictive Control and Neural Network for Implementation

[1] Sudhir Sawarkar, [2] Sonali V Deshmukh

[1] Department of Electronics and Telecommunication
Datta Meghe College of Engineering
Sector 3,Airoli,Navi Mumbai.

[2] Department of Electronics and Telecommunication
Datta Meghe College of Engineering
Sector 3,Airoli,Navi Mumbai.

Abstract - In this paper constrain model predictive control is studied. The discrete time neural network for solving quadratic programming problem is stated. To solve the issue of continuous time neural network, simplified dual neural network is implemented in discrete time neural network. The convergence of discrete time neural network is analyse with the help of a software platform. The system response is obtained from air separation unit.

Keywords - *Model predictive control (MPC), neural network, discrete time neural network, simplified dual neural network (SDNN).*

1. Introduction

The last three decades have seen a rapidly increasing number of applications where control techniques based on dynamic optimisation led to improved performance [195], e.g. maximising the process output or minimising energy use and emissions. These techniques use a mathematical model in the form of differential equations of the process to be controlled in order to predict its future behaviour and to calculate optimised control actions. Sometimes this can be done once before the runtime of the process to yield an offline controller. However, unknown or unmodelled disturbances often call for a feedback controller that repeatedly solves optimal control problems in real-time, i.e., during the runtime of the process. The notion model predictive control (MPC) refers to this second case. Large engineering systems typically consist of a number of subsystems that interact with each other as a result of material, energy and information flows. A high performance control technology such as model predictive control (MPC) is employed for control of these subsystems. Local models and objectives are selected for each individual subsystem. The interactions among the subsystems are ignored during controller design. In plants where the subsystems interact weakly, local feedback action provided by these subsystem (decentralized) controllers may be sufficient to overcome the effect of interactions. For such cases, a decentralized control strategy is expected to work adequately. For many plants, ignoring the interactions among subsystems leads to a significant loss in control performance. An excellent illustration of the hazards of such a decentralized control structure was the failure of the North American power system resulting in the blackout of August 14, 2003.

The decentralized control structure prevented the interconnected control areas from taking emergency control actions such as selective load shedding. As each subsystem tripped, the overloading of the remaining subsystems became progressively more severe, leading finally to the blackout. MPC has a couple of advantages over traditional control approaches. Its key feature is that the control objective as well as desired limitations on the process behaviour can directly be specified within an optimal control problem. MPC naturally handles processes with multiple inputs or outputs as it conceptually can be used with dynamic models of any dimension. These advantages come at the expense that optimal control problems need to be solved in real-time— possibly on embedded controller hardware.

The goal of our paper is to compare constrained MPC [11] with MPC with neural network with the help of various graph. This paper is organized as follows. Section II introducesthe background of MPC technology. In Section III, the convergence property of the discrete-time SDNN is investigated, and a sufficient conditionfor global convergence for discrete-time SDNN is obtained. Section IV applies the proposed design to an ASU system to verify its effectiveness. Section V gives the future scope. Section VI concludes this paper.

2. Model Predictive Control

2.1 Literature

In MPC literature**,** there are many methods which are used for practical application in various industries. The various methods are: linear quadratic Gaussian(LQG),Identification and Command (IDCOM), which is solution software and referred as model predictive heuristic control (MPHC),dynamic matrix control(DMC),quadratic dynamic matrix control(QDMC) [5] ,shell multivariable optimizing controller(SMOC). Each individual method its own issues like some methods requires kalman filter for obtaining system equation, online multiplications etc. Out of these methods DMC is one of the most simple to implement and understand. The advantages of DMC are as plant make use of linear step response model, future output behaviour is specified, have quadratic performance objective over a finite prediction horizon. Each earlier method is cause of next modified version .

To overcome the issues of earlier method Next method will be come into consideration. The earliest MPC method is DMC-plus and RMPCT. The DMC is under the development, so as refer as DMC-plus. The RMPC algorithm offered by Honeywell was merged with the Profimatics PCT controller to create their current offering called RMPCT. DMC-plus and RMPCT are representative of the fourth generation MPC technology. There exists a great variety of different model types within the MPC context. They can be roughly divided into first principles (or white-box) models, identified (or black-box) models, or combinations of both (often called grey-box models). First principles models try to replicate, e.g., physical or chemical laws of nature, whereas identified models are based on measurements of the real process.

2.2. Model Predictive Control

The model predictive control *(MPC)* is an important advanced control technique for difficult multivariable control problems. If a reasonably accurate dynamic model of the process is available, model and current measurements can be used to predict future values of the outputs [2]. Then the appropriate changes in the input variables can be calculated based on both predictions and measurements. In essence, the changes in the individual input variables are coordinated after considering the input-output relationships represented by the process model. In MPC applications, the output variables are also referred to as controlled variables or *CVs*, while the input variables are also called manipulated variables or *MVs*. Measured disturbance variables are called *DVs* or feed forward variables. In view of its remarkable success, MPC has been a popular subject for academic and industrial research .Major extensions of the early MPC methodology have been developed, and theoretical analysis has provided insight into the strengths and weaknesses of MPC.

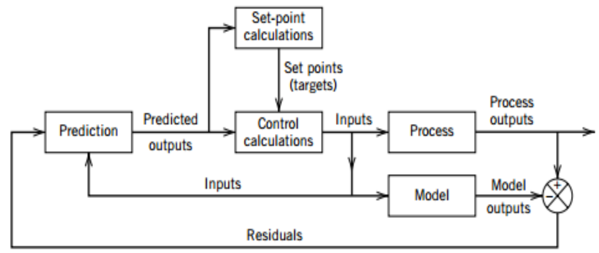


Fig 1: Block diagram for model predictive control

A block diagram of a model predictive control system is shown in Fig 1. A process model is used to predict the current values of the output variables. The residuals, the differences between the actual and predicted outputs, serve as the feedback signal to a Prediction block. The predictions are used in two types of MPC calculations that are performed at each sampling instant: set-point calculations and control calculations. In equality constraints on the input and output variables, such as upper and lower limits, can be included in either type of calculation.

Model predictive control(MPC) repeatedly calculates control actions which optimise the forecasted process behaviour [15]. The prediction is based on a dynamic model of the process to be controlled [6]. At each sampling instant, this leads to an optimal control problem which needs to be solved online. Afterwards, the optimised control action is applied to the process until the next sampling instant when an updated optimal control problem, incorporating the new process state, is solved. Hence, model predictive control is a feedback control strategy, sometimes also referred to as receding horizon control. Unless otherwise stated, we consider time-continuous process models described by an ordinary differential equation(ODE) and an output function. We assume the process model to be defined over a time interval T def= $[t_{start}, t_{end}] \epsilon$ R, and to have the following form:

$$x\,(t_{start}) = w0, \qquad (1)$$
$$\dot{x}\,(t) = f\,(t, x(t), u\,(t), p), \qquad (2)$$
$$y\,(t) = h\,(t, x\,(t), u\,(t), p), \qquad (3)$$

With differential states x:T $\rightarrow R^{nx}$, control inputs (or manipulated variables) u: T $\rightarrow R^{nu}$, time-constant parameters $p\ \epsilon R^{npa}$, and outputs, or controlled variables, y:T$\rightarrow R^{ny}$. Moreover, $w0\ \epsilon R^{nx}$ is the initial value of the differential states, f:D^f =R × R^{nx} × R^{nu} × $R^{npa}$$\rightarrow$R$n$x the right-hand function of the ODE, and h:D^h=R× R^{nx} × R^{nu} × R^{npa} $\rightarrow R^{ny}$ denotes the output functionof the process model.The capability to enforce constraints on the control inputs, outputs or parameters is one of the most important features of MPC. Also additional equality constraints can be expressed using the formulation method. This type of method will be referred as Constrained MPC. Having solved OCP ($t^{(i)}$, $w0$), the optimal control input $u^{opt}(t)$ is applied to the process until the next sampling instant $t^{(i+1)}$. Then the current process state is obtained (measured or estimated) and the optimal control problem OCP ($t^{(i+1)}$, $w0$) is solved with this updated initial value for the process state. This yields the model predictive control scheme which is stated in algorithm 1.1

Algorithm 1 (model predictive control concept)

Input: (open-loop) optimal control problem OCP ($t^{(0)}$, $w0$), sequence of sampling instants $t^{(0)}$, $t^{(1)}$,$t^{(n\text{sample}-1)}$

Output: piecewise defined optimised process inputs u^*: [$t$start, $t$end] $\rightarrow \mathbb{R}^{nu}$

(1) Set $i \leftarrow 0$.

(2) Obtain process state $w0$ at time $t^{(i)}$ and formulate OCP ($t^{(i),}$ $w0$).

(3) Obtain $u^{opt}(t)$, $t \in [t^{(i)}, t^{(i)} + t\text{p}]$, by solving OCP($t^{(i)}$,$w0$).

(4) Set $u^*(t)$ def= $u^{opt}(t)$ $\forall t \in [t^{(i)}, t^{(i+1)}]$ and apply $u^*(t)$ to the process until $t^{(i+1)}$.

(5) Stop if $i = n_{\text{sample}} - 1$, otherwise set $i \leftarrow i + 1$ and continue with step (2).

The conditions which are stated in algorithm 1 are never satisfied in a real-world setup: there are always discrepancies between the model and real process, known as model-plant mismatch, as the real process is too complex to be modelled exactly. Sometimes the process dynamics are not even known completely making approximations or interpolations necessary. Moreover, unknown disturbances are almost always present in real-world and measurement noise[1] impedes the exact determination of the initial process state. On the other hand, the calculated optimal control inputs often cannot be applied exactly to the real process.

2.3. QP Solver

While implementing the MPC algorithm QP solver is one of the important thing of more focusing. There are a lot of method available for solving QP problem of constrained MPC system.We can formulate the optimization problem of theDMC controller as the following standard QP problem:

$$\text{Min}_{\Delta U(k)} J(k) \text{ s.t. } C\Delta U(k) \leq b \qquad (4)$$

Where$J(k) = 1/2\Delta U^T(k)\,(A^TQA + R)\,\Delta U(k) + (\overline{Y_0}(k) - \text{Ref}(k))^TQA\Delta U(k)$. By solving the QP problem (4), the DMC controller can give the control input

$$U(k) = u(k-1) + \Delta u(k/k). \qquad (5)$$

Some of them are active set method [7], interior point method [7], the recurrent neural network method, and the multiparametric method. The multiparametric method is not in used as it is implemented offline. So the real time engineers did not prefer this method as it operates offline and require much more time for computing system response. The active set method and interior point method are operated online. The active set and interior point method are well suited for small scale and large scale application respectively. These methods are widely implemented in variety of applications today. Though both methods are easy, one of major issue is online matrix inversion requirement. Because of it the system response get slower and slower. So there will another choice for overcome this issue is to explicit MPC method. One of the most accurate solution is simplified dual neural network(SDNN).SDNN has the advantages as its grate adaptability, least number of neurons, low network complexity as compared others solutions of explicit MPC method. SDNN [10] works in two modes as continuous time and discrete time .Out of both we are using discrete time model as the system output and future plant behaviour should define earlier. So by using discrete time model we can record the previous system behaviour and can use it as initial response for future operation, which is very difficult to do in continuous time model.

3. Discrete-Time SDNN

The role of neural network in MPC is to classify response at different time interval. We are using dual neural network for better result analysis[8,9]. The SDNN will be discretises from continuous time SDNN [1] by replacing certain terms in discrete time formats

$$v(n+1) - v(n) = \alpha(-EW^{-1}(E^T v(n) - c) + g(EW^{-1}\ (E^T v(n) - c) - \beta v(n))) \ (6)$$

Where$\alpha = T/\epsilon$.The output equations is given by

$$x(n) = W^{-1}E^Tv(n) - W^{-1}c. (7)$$

Equation (d) is the neuron function of the discrete-time SDNN. From the dimension of matrix $E^{W-1}E$, the number of neurons, i.e., the dimension of the state variable $v(n)$, is obviously equal to the number of inequality constraints. Since the function and the performance of the neural network are mainly determined by the neurons, it indicates that the complexity of the proposed method is mainly affected by the number of inequality constraints.

The discrete-time SDNN given by globally converges to the equilibrium point $v*$if

$$0 < \alpha < 2/(\beta + \lambda max.) \qquad (8)$$

λmax denote the largest eigenvalue of $EW^{-1}E^T$.The required convergence will be met if and only if SDNN satisfied the following equation

$$0 = -EW^{-1}(E^T v^* - c) + g(EW^{-1}(E^T v^* - c) - \beta v^*) \qquad (9)$$

However it's very difficult to satisfy this condition in real time environment. It can be made possible by introducing a positive integer which is referred as tolerance. The value of the tolerance should be set to trade-off between accuracy and convergence rate according to the practical requirement. If high accuracy is required, then the tolerance should be set to a rather small value; on the other hand, for a system with a fast sample rate, a larger tolerance to obtain a fast convergence rate is preferred. For the hardware platform, the original SDNN was proposed as a continuous-time model in [1] and was simulated on MATLAB there.

4. Result and Discussion

As stated earlier we are comparing constrained MPC with Neural network employed MPC.The constrained MPC is also referred as general MPC.MPC will done all the mathematical operations in matrix form, we analyse both the system output and system response at different time intervals. Fig 2 shows the system response for constrained MPC as illustrated in algorithm 1.1.We have chosen four time intervals for better understanding the response. y1 is the system response when no previous input is present. y1became stable from 12sec -35sec at the value of 1.5 andthen goes on decaying slightly. For next response y1 willtreated aspreviousor reference input and MPC will

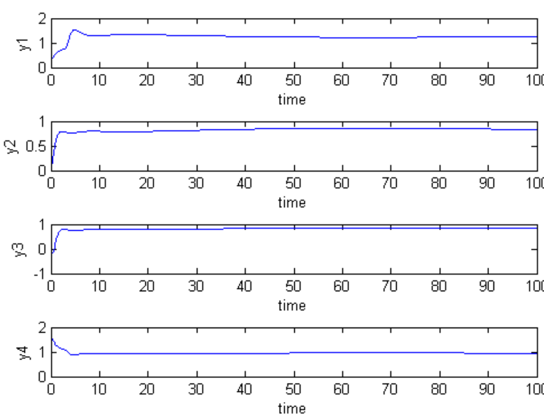


Fig 2: System Response of constrain MPC)

calculate the present response, now its y2, coordinating with present input and reference input. At some again 0.8 value. Similarly present response will act as reference input for next time interval and accordingly y3 and y4 get stable at 0.9 and 1 value as shown in fig 2.response will be at steady state value. For y2 system takes some time to converge, i.e. up to 45 sec and get stable. As we have seen the algorithm, the manipulated variable is introduced for decision making with the help of feedback .Fig 3 shows the variation of manipulated variable(u)with respect to time. For initial values manipulated variable will change continuously, i.e. u1.Once system give the initial response the manipulated variable will change slightly. as shown for u2 and u3which are near to zero. Fig 4 is the important graph which shows theoretical difference between constrain MPC and MPC with neural network.We have consider four plot (y1, y2, y3, y4) for both type

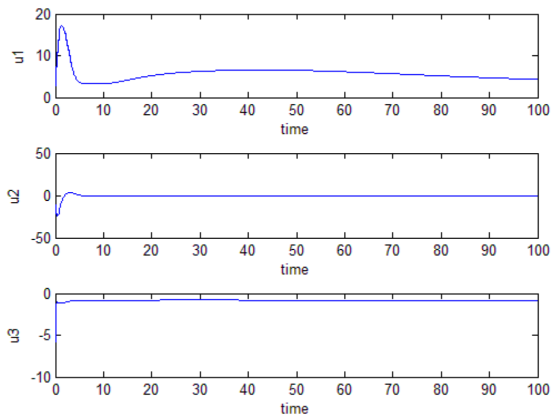


Fig 3. Manipulated variable of constrain MPC

of MPC combinely. In each plot red line indicates response of constrained MPC and blue line indicates response of MPC with neural network. Initially the operation of both MPC methods is same as stated earlier. From y1 we can say the system response will be higher in the time interval 0-25 sec with value as 100 which is much higher than other methods. The difference between actual value of both method is 9-10% of highest value.y2 and y4 are operate same as to the y1.The actual difference we can see from y3.If we carefully observe the y3 plot up to 10sec time period ,we get 10% difference between both the response. In y3 the value of the response is 10 and 20 for constrain MPC and neural network MPC respectively. The minimization operation will try to steer the outputs back into the permitted zone and then act as soft constraints. Fig 5 shows the different input trajectories with respect to ASU system. These input trajectories are consider as reference for present state calculation and get optimal control output. ASU system is consider for getting optimal response. These all trajectories are applied individually at particular time instant. Each signal can be treated separately.

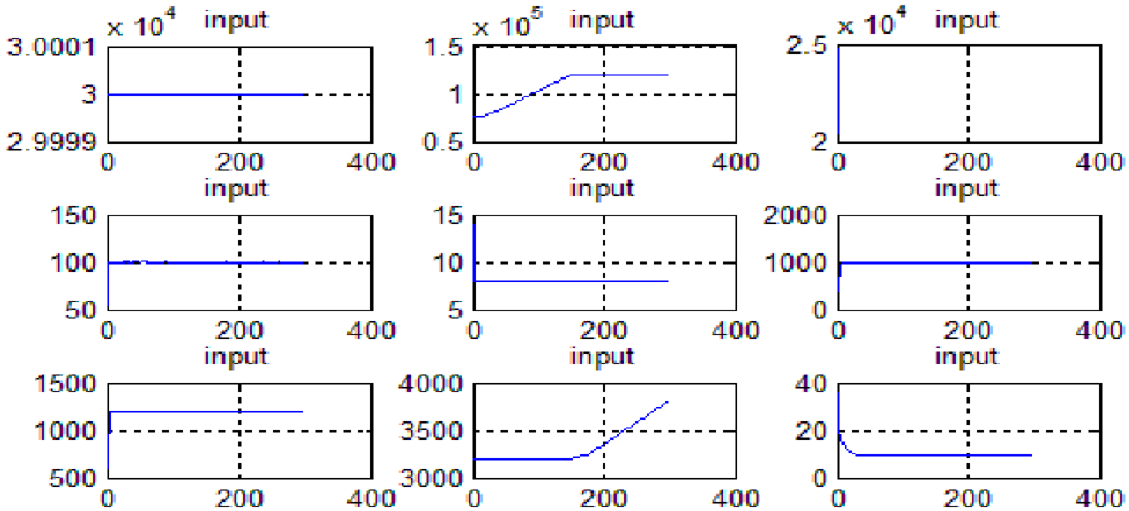
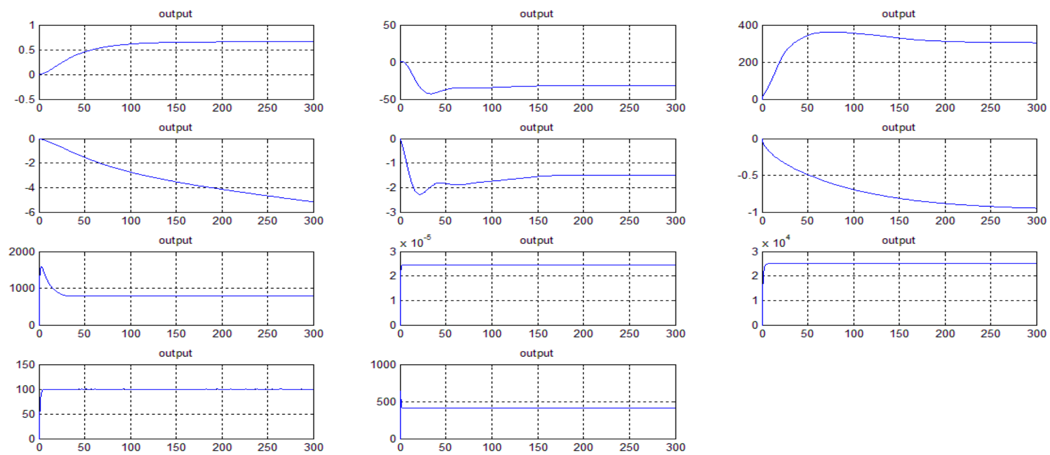


Fig. 5. Input trajectories of the ASU system (horizontal coordinate unit: min).



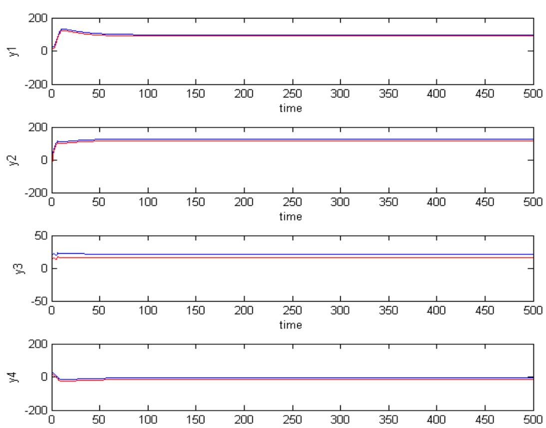


Fig 4. System response of MPC with neural network

The combine response can be obtained on single input trajectory also. These all input trajectories can be implemented on a single platform. The platform can be hardware based or software based. Fig 6 show output trajectories which are obtained by considering all input trajectories.

5. Future Scope

As we have seen that neural network is one of the best implementation method for optimal control. In this paper we coordinate neural network with linear MPC.It can be possible to make use of nonlinear MPC method also .As distributed MPC is one the most trending method in various automation industries. So for getting accurate optimal control neural network can be implement through distributed MPC method.

6. Conclusion

From the results we got we can conclude that model predictive control with neural network is one of the best option if response time is major factor of measurement. This method is widely implemented in automation industries. The neural network is used to classify various signals to respond accurately.

References

[1] W. Greenwell and A. Vahidi, “Predictive control of voltage and current in a fuel cell-ultracapacitor hybrid,” *IEEE Trans. Ind. Electron.* vol. 57, no. 6, pp. 1954–1963, Jun. 2010.

[2] C. Lim, N. A. Rahim, W. Hew, and E. Levi, “Model predictive control of a two-motor drive with five-leg-inverter supply,” *IEEE Trans. Ind. Electron.*, vol. 60, no. 1, pp. 54–65, Jan. 2013

[3] A. Gupta, S. Bhartiya, and P. S. V. Nataraj, “A novel approach to multiparametric quadratic programming,” *Automatica*, vol. 47, no. 9, pp. 2112–2117, Sep. 2011.

[4] L. G. Bleris, P. D. Vouzis, M. G. Arnold, and M. V. Kothare, “A coprocessor FPGA platform for the implementation of real-time model predictive control,” in *Proc. Amer. Control Conf.*, 2006, pp. 1912–1917.

[5] D. Li, N. Yang, R. Niu, H. Qiu, and Y. Xi, “FPGA based QDMC control for reverse-osmosis water desalination system,” *Desalination*, vol. 285, pp. 83–90, Jan. 2012.

[6] N. Yang, D. Li, J. Zhang, and Y. Xi, “Model predictive controller design and implementation on FPGA with application to motor servo system,” *Control Eng. Pract.*, vol. 20, no. 11, pp. 1229–1235, Nov. 2012.

[7] M. S. K. Lau, S. P. Yue, K. V. Ling, and J.M.Maciejowski, “A comparison of interior point and active set methods for FPGA implementation of model predictive control,” in *Proc. Eur. Control Conf.*, 2009, pp. 156–161.

[8] S. Zhang and A. G. Constantinides, “Lagrange programming neural networks,” *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, Vol 39, no.7, pp 441-452, July 1992

[9] M. Yashtini and A. Malek, “A discrete-time neural network for solving nonlinear convex problems with hybrid constraints,” *Appl. Math. Compute*vol. 195, no. 2, pp. 576–584, Feb. 2008.

[10] Y. Pan and J. Wang, “Model predictive control of unknown nonlinear dynamical systems based on recurrent neural networks,” *IEEE Trans. Ind. Electron.*, vol. 59, no. 8, pp. 3089–3101, Aug. 2012

[11] J. B. Rawlings and K. R. Muske, “The stability of constrained receding horizon control,” *IEEE Trans. Autom. Control*, vol. 38, no. 10, pp. 1512– 1516, Oct. 1993.

[12] Z. Xu *et al.*, “Automatic load change system of cryogenic air separation process,” *Sep. Purif. Technol.*, vol. 81, no. 3, pp. 451–465, Oct. 2011.

[13] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert, “Constrained model predictive control: Stability and optimality,” *Automatica*, vol. 36, no. 6, pp. 789–814, Jun. 2000.

[14] D. Kinderlehrer and G. Stampacchia, *an Introduction to Variational Inequalities and Their Applications*. New York, NY, USA: Academic, 1980.*TMS320C6678 Multicore Fixed and Floating-Point Digital Signal Processor Data Manual*, Texas Instruments Incorporated, Dallas, TX, USA, 2011.

[15] AllgoK wer, F., & Zheng, A. (1999). *Model predictive control*: *Assessment and future directions*. Proceedings of international workshop onmodel predictive control, Ascona, 1998, Berlin: Springer.*23*(2), 149}160.

[16] Badgwell, T. A. (1997). Robust model predictive control ofstable linearsystems. *International Journal of Control*, *68*(4), 797},818.

[17] Bemporad,A,Casavola A and Mosca.E(1997).Nonlinear control of constrained linear system via predictive reference management,IEEE Transactions on Automatic Control,42(3),340]349.

[18] Clarke.D.W,Mohtadi.C and Tus.P.S(1987b).Generalize predictive control.Part 2:Extansions and interpretations.Automatic.

[19] Chisci, L., Lombardi, A., & Mosca, E. (1996). Dual receding horizon control of constrained discrete-time systems. *European Journal ofControl*, *2*, 278}285.

[20] Chen, C. C., & Shaw, L. (1982). On receding horizon feedback control. *Automatica*, *18*, 349}352.

[21] GarcmHa,C.E&Morshedi,A.M(1986).Quadratic programming solution of dynamic matrix control(QDMC).Chemical Engineering communications,46,73}87.

Sudhir Sawarkar receives Ph.D. in computer engineering from Amravati University in 2007.Currently he is principal of Datta Meghe College of Engineering. He has 27 years of teaching experience. He has published 37 and 47 papers in international conference and journals .he has published 8 papers in national conferences. He is member of IndianSociety for Technical Education, Institute of technical engineering, Computer society of India and Board of studies of Computer engineering.

Sonali Deshmukh receives B.E in electronics engineering from Mumbai University in 2012. She currently pursuing M.E in electronics engineering from Mumbai University. She has 1 year of teaching experience. She has published a paper in International conference.