Adaptive Control of Self-Balancing Two-wheeled Robot System based on Online Model Estimation

Haris Sheh Zad, Abasin Ulasyar

Department of Electrical and Electronics Engineering
Koç University, Istanbul, Turkey

Abstract

In this article, an adaptive model predictive controller (MPC) is designed for the position control of the self-balancing two-wheeled robot system. The system future output is optimized using the MPC controller by computing the manipulated variable trajectory. Traditional MPC uses a Linear-Time-Invariant (LTI) dynamic model of the system for the prediction of future behavior. The model of the self-balancing two-wheeled robot system is strongly nonlinear which degrades the prediction accuracy of the traditional MPC controller. Therefore, an adaptive MPC controller is designed based on linear-time-varying Kalman filter which online tunes and updates the estimated system parameters and accordingly produces the control effort in the presence of the input/output and state constraints. The performance of the proposed controller is compared with the traditional MPC controller and PID controller. The results show improved reference tracking and better stability for the proposed adaptive MPC controller as compared to traditional MPC and PID controller.

1. Introduction

An inverted pendulum is used in order to approximate the model of two-wheeled robot system [1]. In the literature, various control schemes have been reported for the position control of the two-wheeled robot system. [2] developed a state-feedback controller for stabalizing thse inverted pendulum. [3] developed a two-wheeled inverted pendulum which had steering ability. [4] designed a mobile robot based on a two-wheeled differential drive inverted pendulum. [5] designed a robot based two-wheeled wheelchair.The robot is balanced by a simple PI control scheme. [6] designed a Linear quadratic Gaussian (LQG) based control scheme in order to control the robotic wheelchair. [7] developed a nonlinear control scheme for regulating the inverted pendulm position. [8] proposed a feedback-error-learning (FEL) controller for the purpose of stablizing a double pendulum. A comparison between PID controller and fuzzy logic controller for stablizing a double inverted pendulum has been presented by [9]. The controller utilized a Pole placement state-feedback technique. A cascade control loop is utilized in order to reduce the complexity of the controller. However, there is a shortcoming with the feedback linearization techniques that they require an exact information regarding the system parameters.

MPC controller is largely utilized in industry in order to design controllers for the nonlinear complex processes [10]. The minimization of the cost function is done for obtaining optimal control for the plant [11, 12]. Regardless of the fact that most of the practical processes have nonlinear nature, MPC mostly addresses the optimization problem by linearizing the nonlinear plant model, because the linear model is fast and easy to develop as compared to nonlinear model. Also, the robustness and stability are relatively easy to provide in linearized model as compared to the nonlinear model. Due to these facts, the nonlinear MPC application is limited but has a great potential [13]. For a strongly nonlinear plant model, the linearization of the system does not give acceptable performance in all the operating conditions. For such systems, satisfactory results cannot be obtained from the linearization of the plant, until and unless the plant always operates in the neighborhood of the operating conditions. Adaptive linearization of a plant online updates the linearizing process based on the measurement data. [14] proposed control strategies based on the multistep Newton-type, in which the linearization of the nonlinear plant is done around a nominal trajectory and then the quadratic problem is solved over the horizon. [15] proposed a dual mode robust controller based on receding horizon for a class of nonlinear systems having model error and control and state constraints. In this paper, a linear time varying Kalman filter is used to online update the gains of the filter at each control interval in order to keep uniformity with the updated model of the process. In section 2, the analytical model of the two-wheeled robot is given. The design of the adaptive MPC controller is provided in section 3. Section 4 consists of the results after applying the designed adaptive MPC controller to the robot system. Section 5 consists of the conclusion.

2. Analytical Model of Two-Wheeled Robot System

The analytical model of two-wheeled robot is obtained using an inverted pendulum fixed on the top of the cart moving in a single plane as depicted in Fig. 1.

By utilizing the Newton's law, the force equations are obtained as follows:

$$m_c \frac{dx^2}{dt^2} = u - Q \tag{1}$$

$$Q = m\frac{d^2}{dt^2}(x + l_c) = m\ddot{x} + ml_c\ddot{\emptyset}\cos\emptyset - ml_c\sin\emptyset \tag{2}$$

$$mg - P = m\frac{d^2}{dt^2}(l_c\cos\emptyset) = ml_c[-\ddot{\emptyset}\sin\emptyset - (\dot{\emptyset})^2\cos\emptyset] \tag{3}$$

$$mgl_c\sin\emptyset = mL\ddot{\emptyset}\,.\,l_c + m\ddot{x}l_c\cos\emptyset \tag{4}$$

By making the assumption that $\emptyset$ and $\dot{\emptyset}$ is very small, the system is linearized and the linearized equations are shown as following:

$$m_c\ddot{}x = u - m\ddot{x} - mL\ddot{\emptyset} \tag{5}$$

$$g\emptyset = l_c + \ddot{x} \tag{6}$$

$$m_c\ddot{x} = u - mg\emptyset \tag{7}$$

$$m_cl_c\ddot{\emptyset} = (m_c + m)g\emptyset - u \tag{8}$$

The Laplace transforms of the above equations are given as follows:

$$m_cs^2\hat{x}(s) = \hat{u}(s) - mg\hat{\emptyset}(s) \tag{9}$$

$$m_c l_c \hat{\emptyset}(s) = (M+m) g \hat{\emptyset}(s) - \hat{u}(s) \quad (10)$$

Therefore,

$$\hat{g}_{xu}(s) = \frac{s^2 - g}{s^2[m_c s^2 - (m_c + m) g]} \quad (11)$$

$$\hat{g}_{\emptyset u}(s) = \frac{-1}{m_c s^2 - (m_c + m) g} \quad (12)$$

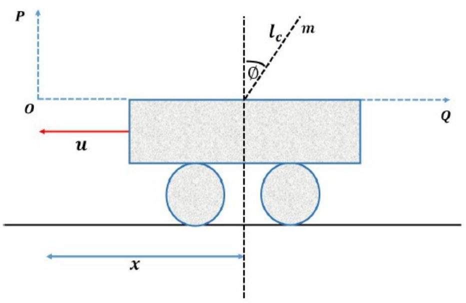

Fig. 1. Inverted pendulum system with the cart

By selecting the state variables as $x_1 = x, x_2 = \dot{x},\ x_3 = \emptyset$ and $x_4 = \dot{\emptyset}$, the state space representation is shown follows:

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\\ \dot{x}_3\\ \dot{x}_4\end{bmatrix} = \begin{bmatrix}0 & 1 & 0 & 0\\ 0 & 0 & -mg/m_c & 0\\ 0 & 0 & 0 & 1\\ 0 & 0 & (m_c+m)g/(m_c l_c) & 0\end{bmatrix}\begin{bmatrix}x_1\\ x_2\\ x_3\\ x_4\end{bmatrix} + \begin{bmatrix}0\\ 1/m_c\\ 0\\ -1/m_c l_c\end{bmatrix} u$$

$$y = [1\ 0\ \ 0\ 0]\begin{bmatrix}x_1\\ x_2\\ x_3\\ x_4\end{bmatrix} \quad (13)$$

The parameters used for the system are; Mass of the cart $m_c = 0.7\ kg$, mass of the pendulum $m = 0.4kg$ and length of the pendulum $l_c = 0.4m$

3. Adaptive Model Predictive Controller Design

Model Predictive Control (MPC) is one of the popular technique of optimization in control. It repeatedly evaluates the control input at each instance of time using a finite horizon problem of optimization based on the given system model. It is the most significant option for handling control problems for systems with constraints on inputs and states. As most of the models in practice are non-linear, MPC is widely implemented on linear models. The reason behind is this, that it is simple and fast to develop a linear model as compared to nonlinear. In terms of stability and robustness, it is hard to achieve these two in nonlinear systems because the nonlinear systems and constraints leads to a non-convex problem of optimization. These issues are difficult to address in nonlinear systems. Therefore, the implementation of MPC for nonlinear systems is very limited in practice as compared to linear systems. For dealing with uncertainties in a system model, a robust MPC works very well. However, the performance of control is degraded by using the fixed model of system.

Using MPC control, there are prediction errors which are quite significant if the plant is highly nonlinear. For this purpose, Adaptive MPC is used so that MPC performance is not degraded due to the prediction errors. The adaptive nature of MPC uses a fixed model in such a way so that its LTI model parameters evolve gradually as the time progress. The Adaptive MPC always updates the model of plant and nominal operating conditions for every control interval and it remains same over the prediction horizon. In order to design the adaptive MPC controller, first a linear optimal MPC controller is designed based on the initial operating conditions and then online estimation method is used in order to obtain the adaptive MPC controller.

The optimal MPC is designed for the self-balancing two wheeled robot system in [16], and is given as follows. The control trajectory is modeled using a set of orthonormal functions. The control signal should have a bound on its integral squared value. If the control signal is represented by $u(t)$, then the subsequent condition should be met as given in [16].

$$\int_0^\infty u(t)^2 dt < \infty$$

In case of constant reference signal, the following condition should meet in order to design the MPC controller.

$$\int_0^\infty \dot{u}(t) dt < \infty$$

The state space equations of the system having r inputs and q outputs is given as below;

$$\dot{x}_r(t) = A_r x_r(t) + B_r u(t)$$
$$y(t) = C_r x_r(t)$$

where, $x_r(t)$ have dimensions of n1. A_r, B_r and C_r have dimensions of $m_1 \times m_1, m_1 \times r$ and $q \times r$ respectively.

By defining the auxiliary variables as following:

$$z(t) = \dot{x}_r(t)$$
$$y(t) = C_r x_r(t)$$

and by choosing new state variables as $x(t) = [z(t)^T\ y(t)^T\ ]^{\mathrm{T}}$, the following state space model is obtained in augmented form:

$$\begin{bmatrix}\dot{z}(t)\\ \dot{y}(t)\end{bmatrix} = \begin{bmatrix}A_r & 0_r^T\\ C_r & 0_{q\times q}\end{bmatrix}\begin{bmatrix}z(t)\\ y(t)\end{bmatrix} + \begin{bmatrix}B_r\\ 0_{q\times r}\end{bmatrix}\dot{u}(t)$$

$$y(t) = [O_r \quad I_{q\times q}]\begin{bmatrix}z(t)\\ y(t)\end{bmatrix} \quad (14)$$

Where $I_{q\times q}$ represent identity matrix, $0_{q\times q}$ and $0_{q\times r}$ represent zero matrices. By rewriting Equation (14):

$$\dot{x}(t) = Ax(t) + B\dot{u}(t)$$
$$y(t) = Cx(t)$$

In case of disturbance, the model is given as follows:

$$\dot{x}_r(t) = A_r x_r(t) + B_r u(t) + B_d w(t)$$
$$y(t) = C_r x_r(t)$$

where w(t) represents unmeasured disturbance.

The cost function is given as follows:

$$J = \int_0^{T_L} (x(t_n + \tau|t_n)^T\ Qx(t_n + \tau|t_n) + \dot{u}(\tau)^T R\dot{u}(\tau)) d\tau \quad (15)$$

The matrices Q and R are positive semidefinite. The cost function in case of MPC controller is selected as follows:

$$J = \int_0^{T_L} ((r(t_n) - y(t_n + \tau|t_n))^T\ (r(t_n) - y(t_n + \tau|t_n)) + \dot{u}(\tau)^T R\dot{u}(\tau)) d\tau$$

In case of a constant reference, the reference is subtracted from $y(t_n + \tau|t_n)$ and the resulting model is given as follows:

$$\begin{bmatrix}\dot{z}(t_n+\tau|t_n)\\ \dot{e}(t_n+\tau|t_n)\end{bmatrix} = \begin{bmatrix}A_r & O_r^T\\ C_r & O_{q\times q}\end{bmatrix}\begin{bmatrix}z(t_n+\tau|t_n)\\ e(t_n+\tau|t_n)\end{bmatrix} + \begin{bmatrix}B_r\\ O_{q\times r}\end{bmatrix}\dot{u}(\tau)$$

Where $e(t_n + \tau|t_n)$ is $y(t_n + \tau|t_n) - r(t_n)$. $r(t_n)$ is denoted by a constant vector having length $0 \le \tau \le T_L$. Q is selected as $C^T C$, and the cost function is selected as given in Equation (15). Choosing R as following:

$$R = diag(r_s)$$

where $s = 1,2,3,..r$

Therefore,

$$\int_0^{T_L} \dot{u}(\tau)^T R\dot{u}(\tau)\, d\tau = \sum_{s=1}^{r} \int_0^{T_L} r_s \dot{u}_s(\tau)^2\, d\tau$$

$$\int_0^{T_L} u_s(\tau)^T u_s(\tau)\, d\tau \approx \int_0^{\infty} \gamma_s{}^T G_s(\tau) G_s(\tau)^T \gamma_s d\tau = \gamma_s{}^T \gamma_s$$

where $\int_0^{\infty} G_s(\tau) G_s(\tau)^T \, d\tau$ is the identity matrix. Hence;

$$J = \int_0^{T_L} (x(t_n + \tau|t_n))^T Q x(t_n + \tau|t_n) d\tau + \gamma_s{}^T R_G \gamma_s \quad (16)$$

Here R_G denotes the block diagonal matrix and the sth block is given as R_s. By rewriting (17):

$$J = \int_0^{T_L} (e^{A\tau} x(t_n) + \vartheta(\tau)^T \gamma)^T Q\, (e^{A\tau} x(t_n) + \vartheta(\tau)^T \gamma) + \gamma_s{}^T R_G \gamma_s \quad (17)$$

Where $(e^{A\tau} x(t_n) + \vartheta(\tau)^T \gamma)$ is the prediction value of $x(t_n + \tau|t_n)$. Equation (17) denotes a quadratic function of γ. Further,

$$J = \gamma^T \left(\int_0^{T_L} \vartheta(\tau) Q \vartheta(\tau)^T d\tau + R_G \right) \gamma + 2\gamma^T \int_0^{T_L} \vartheta(\tau) Q e^{A\tau} d\tau x(t_n) + x(t_n)^T \int_0^{T_L} e^{A^T\tau} Q e^{A\tau} d\tau\, x(t_n) \quad (18)$$

Letting $\int_0^{T_L} \vartheta(\tau) Q \vartheta(\tau)^T d\tau + R_G = \mu$ and $\int_0^{T_L} \vartheta(\tau) Q e^{A\tau} d\tau = \rho$ and by the square completion of (18), the optimal value of γ becomes as following:

$$\gamma = -\mu^{-1} \rho x(t_n)$$

And the minimum value of the cost function comes out to be,

$$J_{min} = x(t_n)^T \left(\int_0^{T_L} e^{A^T\tau} Q\, e^{A\tau} d\tau - \rho^T \mu^{-1} \rho \right) x(t_n) \quad (19)$$

The Adaptive MPC always updates the model of plant at nominal operating conditions for every control interval and it remains same over the prediction horizon.

The discrete-time LTI model of plant is used as the basis for adaptive MPC. The model is given as below:

$$x(n+1) = Ax(n) + B_u u(n) + B_v v(n) + B_d d(n) \quad (20a)$$
$$y(n) = Cx(n) + D_v v(n) + D_d d(n) \quad (20b)$$

Where n denotes discrete time index, x are m_x states of plant model, u are n_u manipulated inputs controlled by MPC controller, v are m_v measured disturbance inputs, d are m_d unmeasured disturbance inputs and y are m_y plant outputs which consist of measured m_{ym} and unmeasured m_{yu} outputs. The m_x, m_u, m_y, m_d, m_{ym} and m_{yu} should be all constants. Furthermore, the configuration for input and output signals should remain constant. There are some other requirements necessary for control of adaptive MPC, that are time delay and sample time (T_s). In addition to this, there should be no direct feed-through component, meaning that $D_u = 0$ for the above discrete-time model.

In adaptive MPC controller as the time passes, the nominal operating point is updated along with the plant model. For the nominal conditions the above discrete time plant model is modified as shown below:

$$x(n+1) = \bar{x} + A(x(n) - \bar{x}) + B(u_t(n) - \overline{u_t}) + \overline{\Delta x} \quad (21a)$$
$$y(n) = \bar{y} + C(x(n) - \bar{x}) + D(u_t(n) - \overline{u_t}) \quad (21b)$$

In the above Eq. (21), A, B, C and D are defined as system's parameter matrices which are to be updated. u_t is designated as the combined plant input variable, consisting of u, v and d variables as given in Eq. (20). There are several conditions imposed for nominal operations of plant such as; $\bar{x}$ are nominal states "n_x", $\overline{\Delta x}$ are increments of nominal state "n_x", $\overline{u_t}$ are nominal inputs "n_{ut}" and $\bar{y}$ are nominal outputs "n_y".

In estimation of states, linear MPC uses a Static Kalman Filter (SKF) for updating the states of a controller. This includes states of plant model n_{xp}, states of disturbance model $n_d(\geq 0)$, and states of measurement noise model are $n_n(\geq 0)$. L and M are the two gain matrices, which are the essential requirement for SKF. Besides this, during the time of initialization, the model predictive controller (MPC) evaluates these gain matrices. These gain matrices truly depend upon the models of plant, disturbance and noise parameters as well as the stochastic and random signals which is vital for the operations of models of noise and disturbance.

The significance of Kalman filter is clearly observed when it is used in Adaptive MPC. It tunes the gain matrices L and M at each and every control interval and maintain perfect synchronism while the plant model is updated simultaneously as shown in Eq. (22), which is a linear time varying Kalman filter.

$$L_n = \left(A_n P_{n|n-1} C_{s,k}^T + N\right)\left(C_{s,n} P_{n|n-1} C_{s,n}^T + R\right)^{-1} \quad (22a)$$
$$M_n = \left(P_{n|n-1} C_{s,n}^T\right)\left(C_{s,n} P_{n|n-1} C_{s,n}^T + R\right)^{-1} \quad (22b)$$
$$P_{n+1|n} = A_n P_{n|n-1} A_n^T - \left(A_n P_{n|n-1} C_{s,n}^T + N\right) L_n^T + Q \quad (22c)$$

In the above Eq. (22), Q, R and N are covariance matrices which are constant in nature as clearly defined in state estimation of MPC.

The parameter matrices of state space are A_n and $C_{s,n}$, these are used in the state of controller as can be seen in MPC. The gain matrices are L_n and M_n, which updates at every discrete time index n. The error of state estimate of a covariance matrix is designated as the value of $P_{n+1|n}$, which is defined at every n index of time.

4. Results

For simulating the design adaptive MPC controller for the position control of Self-balancing robot system, Matlab is used as simulation tool. A discrete time Self-balancing robot system model is online identified at each control interval and then the internal plant model is updated using the adaptive MPC controller achieving successfully the nonlinear control. The single wheeled robot system nonlinear model is first linearized at the initial operating conditions.

A linear optimal MPC controller is designed first at the initial operating conditions and then it is combined with the online estimation scheme in order to implement the adaptive MPC scheme. The constraints of -1.5 and +1.5 are imposed on the manipulated variable and the corresponding response of the system is displayed in Fig. 2. The system is tracking the reference position quite well in the presence of the constraints.

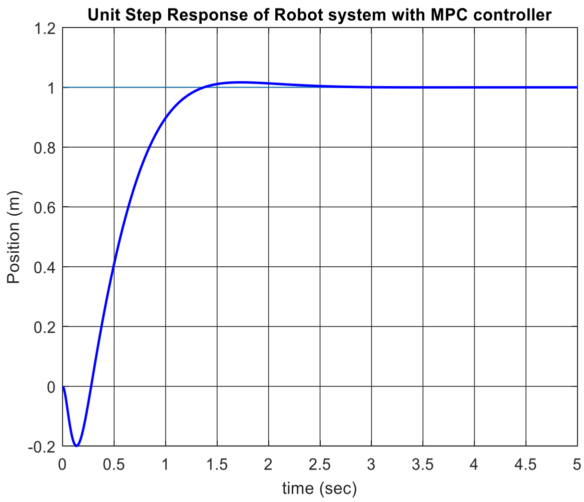

Fig. 2. MPC controller response with constraints on the variables

A comparison is established between the MPC controller and the traditional PID controller for the reference tracking of the self-balancing robot. Fig. 3 shows the corresponding comparison. The tuning of the PID controller is performed by using the Ziegler-Nichols method.

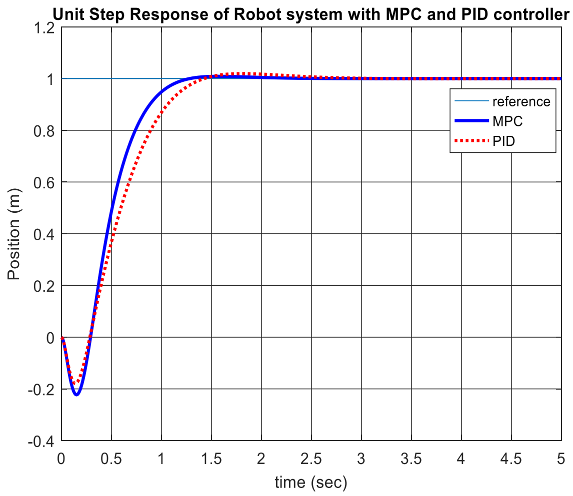

Fig. 3. Comparison of the PID and MPC controller

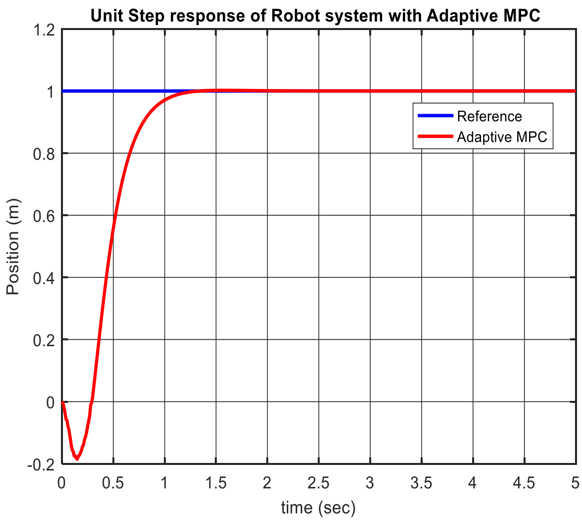

Fig. 4. Unit Step response of robot system with adaptive MPC

The online estimation is performed using Kalman filter with the noise covariance of 0.012. Fig. 4 shows the results of applying adaptive MPC controller to the self-balancing robot system for its position control.

The system has a settling time of 0.95 sec. There is no steady state error present in the system. Fig. 5 shows the comparison of adaptive MPC and non-adaptive MPC controllers for the position control of the self-balancing robot system. The negative overshoot has decreased in the adaptive MPC scheme, also the settling time has reduced significantly as compared to the non-adaptive MPC controller.

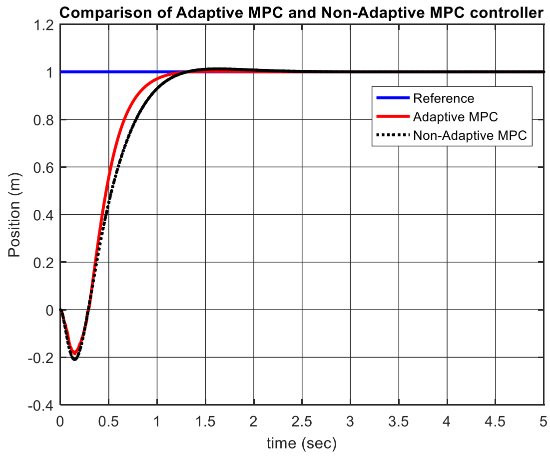

Fig. 5. Comparison of adaptive and non-adaptive MPC controller

5. Conclusion

Self-balancing two wheeled robot system is an adaptive system mimicing human nature in avoiding the obstacles while moving. In this article, an adaptive Model Predictive Controller is designed in order to track the position of Self-balancing Two-wheeled robot. Matlab is utilized in order to simulate the designed adaptive MPC controller. Step reference tracking is used in order to evaluate the performacne of the designed adaptive controller. Results indicate that MPC controller performs better in the presence of manipulated variable constriants, nonlinearities, and perturbations in the system dynamics. A comparison is estabished amongst the responses of the adaptive MPC controller and non-adaptive MPC controller. The results indicate that tracking accuracy and ste-speed for the adaptive MPC controller are better as compared to the non-adaptive MPC controller.

6. References

[1] A. M. Almeshal, K. M. Goher, and M. O. Tokhi, "Dynamic modelling and stabilization of a new configuration of two-wheeled machines," *Robotics and Autonomous Systems,* vol. 61, no. 5, pp. 443-472, 2013.

[2] Z. Lin, A. Saberi, M. Gutmann, and Y. Shamash, "Linear controller for an inverted pendulum having restricted travel-a high-and-low gain approach," in *American Control Conference, Proceedings of the 1995*, 1995, vol. 5, pp. 2980-2984: IEEE.

[3] M. Fiacchini *et al.*, "Design and experimentation of a personal pendulum vehicle," in *Proc. Seventh Portuguese Conf. Automatic Control*, 2006.

[4] J. Li, X. Gao, Q. Huang, Q. Du, and X. Duan, "Mechanical design and dynamic modeling of a two-wheeled inverted pendulum mobile robot," in *Automation and Logistics, 2007 IEEE International Conference on*, 2007, pp. 1614-1619: IEEE.

[5] Y. Takahashi, S. Ogawa, and S. Machida, "Front wheel raising and inverse pendulum control of power assist wheel chair robot," in *Industrial Electronics Society, 1999. IECON'99 Proceedings. The 25th Annual Conference of the IEEE*, 1999, vol. 2, pp. 668-673: IEEE.

[6] Y. Takahashi and O. Tsubouchi, "Modern control approach for robotic wheelchair with inverse pendulum control," in *Intelligent Systems Design and Applications, 2005. ISDA'05. Proceedings. 5th International Conference on*, 2005, pp. 364-369: IEEE.

[7] C. C. Chung and J. Hauser, "Nonlinear control of a swinging pendulum," *Automatica,* vol. 31, no. 6, pp. 851-862, 1995.

[8] E. Kiankhah and M. Teshnelab, "Feedback-error-learning for stability of double inverted pendulum," in *Systems, Man and Cybernetics, 2009. SMC 2009. IEEE International Conference on*, 2009, pp. 4496-4501: IEEE.

[9] M. Tinkir, M. Kalyoncu, U. Onen, and F. M. Botsali, "PID and interval type-2 fuzzy logic control of double inverted pendulum system," in *Computer and Automation Engineering (ICCAE), 2010 The 2nd International Conference on*, 2010, vol. 1, pp. 117-121: IEEE.

[10] J. B. Rawlings, "Tutorial overview of model predictive control," *IEEE Control Systems,* vol. 20, no. 3, pp. 38-52, 2000.

[11] A. Ulasyar and H. S. Zad, "Robust & optimal model predictive controller design for twin rotor MIMO system," in *"2015 9th International Conference on Electrical and Electronics Engineering (ELECO)"*, 2015, pp. 854-858: IEEE.

[12] H. S. Zad, A. Ulasyar, and A. Zohaib, "Robust Model Predictive position Control of direct drive electro-hydraulic servo system, in *"2016 International Conference on Intelligent Systems Engineering (ICISE)"*, 2016, pp. 100-104: IEEE.

[13] E. F. Camacho and C. B. Alba, *Model predictive control*. Springer Science & Business Media, 2013.

[14] W. C. Li and L. T. Biegler, "A multistep, Newton-type control strategy for constrained, nonlinear processes," in *American Control Conference, 1989*, 1989, pp. 1526-1527: IEEE.

[15] H. Michalska and D. Q. Mayne, "Robust receding horizon control of constrained nonlinear systems," *IEEE Transactions on automatic control,* vol. 38, no. 11, pp. 1623-1633, 1993.

[16] H. S. Zad, A. Ulasyar, A. Zohaib, & S. S. Hussain, Optimal Controller Design for Self-Balancing Two-Wheeled Robot System, " *2016 International Conference on Frontiers of Information Technology (FIT)",* (pp. 11-16), IEEE.