International Journal of Innovative Technology and Interdisciplinary Sciences
www.IJITIS.org
ISSN: 2613-7305
Volume 2, Issue 1, pp. 62-77, 2018
DOI: https://doi.org/10.15157/IJITIS.2018.2.1.62-77
Received September 15, 2018; Accepted November 14, 2018

Time Variant Predictive Control of Autonomous Vehicles

John Pumwa

Department of Mechanical Engineering,
Papua New Guinea University of Technology (UNITECH), Papua New Guinea
pumwa@mech.unitech.ac.pg

ABSTRACT

This paper develops a linearized time variant model predictive control (MPC) approach for controlling autonomous vehicle tracking on feasible trajectories generated from the vehicle nonlinear ordinary differential equations (ODEs). The paper is an application of the results from computational schemes for nonlinear model predictive control published in (*International Journal of Control, Automation and Systems* 2011 9(5), 958-965; *Mechatronics 2013*, Trajectory Generation for Autonomous Vehicles, 615-626, Springer). The vehicle nonlinear dynamic equations are derived and solved in MPC optimizer. Solution for the closed loop control is obtained by solving online the vehicle dynamic ODEs. Simulations for the new schemes are presented and analysed.

Keywords: Linearized time variant, model predictive control, nonlinear ordinary differential equations, tracking trajectory, trajectory generation.

1. INTRODUCTION

Autonomous vehicles have been received considerable attention in recent years and the need is arising for the new systems that can be able to drive the vehicle automatically from any start points to any destination points generated from the global positioning system (GPS) maps and subject to the vehicle physical constraints. This paper develops model predictive control (MPC) scheme for controlling autonomous vehicle tracking reference setpoints generated online using solver for its ordinary differential equations (ODEs).

Motivation for the use of MPC is its ability to handle the constraints online within its open-loop optimal control problems while most of other control techniques are unfavorable in handling the online constraints or even try to avoid them, thus, losing the best achievable performance. MPC can calculate the real-time optimal inputs and make the close loop system operating near the constrained limits and hence, yield much better performances.

To deal with the system uncertainties and the model-plant mismatches, some robust MPC schemes are being developed accounting for the modeling errors at the controller design. Robust MPC can forecast all possible models in the plant uncertainty set and then, the optimal action can be determined through the min-max optimization. Schemes for robust MPC tracking setpoints can be referred to from Minh V.T and Hashim F.B (2011) [1], where the system's uncertainties are represented by a set of multiple models via a tree trajectory and its branches, and the robust MPC problem is to find the optimal control actions that, once implemented, cause all branches to converge to a robust control invariant set. Other MPC algorithms for controlling vehicle speed and engine torque are referred to from Minh V.T and Hashim F.B (2012) [2], where a real time

transition strategy with MPC is developed for achieving quick and smooth clutch engagements for hybrid electrical vehicles. Study on vehicle handling and steering system is referred to in Minh V.T (2012), chapter 8 [3], where the vehicle dynamic behaviors are analyzed and used to design a fee-error feedback controller for its autonomous tracking.

Robust MPC schemes for intput saturated and softened state constraints can be referred to from Minh V.T and Afzulpurkar N (2005) [4], where uncertain systems are designed with the use of linear matrix inequalities (LMIs) subject to input and output saturated constraints. Nonlinear MPC (NMPC) calculations are referred to in Minh V.T and Afzulpurkar N (2006) [5], where three NMPC schemes with zero terminal region, quasi-infinite horizon, and softened state constraints are presented and compared. In these NMPC schemes, all online inputs solution from the NMPC optimizer is implemented for the close-loop control by solving on-line the ODEs repeatedly.

Control of tracking vehicle with MPC can be referred to recently in several research papers. However the idea of using MPC for tracking trajectories generated online from ODEs with time variant system is still missing. An MPC scheme for autonomous ground vehicle can be read in Falcon P. et al (2008) [6], where an initial frame work based on MPC is presented. However, this paper fails to mention the real-time solving of the vehicle ODEs equations and the calculation of the optimal controlled inputs for the vehicle velocity and its steering velocity. Similarly, another paper on MPC for path tracking of autonomous vehicles is presented by Lei L. et al (2011) [7], where the vehicle equations of motion are approximately linearized from the vehicle coordinates and the heading angle. However the paper fails to include the steering angle into the dynamic equations.

A robust MPC for mobile vehicle trajectory control can be read in Baharonian M. el al (2011) [8], this paper comes out with an assumption that there is already a virtual reference trajectory and then, the control problem becomes too simple and trivial. An adaptive trajectory tracking control of wheeled mobile is developed by Wang J. et al (2011) [9], however this paper does not mention on how a feasible trajectory can be generated and how optimal control actions can be achieved for the best tracking performance. Another article by Shim T. et al (2012) [10] derives a NMPC to control the front steering velocity and the wheel torque for an autonomous ground vehicle, however, the paper fails to apply the on-line solving ODEs for this NMPC. MPC application for automotive clutch and vibration control can be referred to in Minh V.T [11]. The tracking trajectory generation for autonomous vehicle is presented in Minh V.T [12].

Therefore, idea of this paper is to develop comprehensive MPC schemes for tracking reference trajectories generated online by ODEs from the vehicle dynamics. The vehicle geographic position data can be updated online the GPS map, and then, feasible trajectory setpoints can be automatically generated subject to vehicle constraints on speed, steering, sideslip, obstacles, etc.. The paper is constructed as follows: section 2 describes the system modeling; section 3 develops MPC schemes; section 4 presents MPC simulations; and finally, some study remarks are concluded in section 5.

2. SYSTEM MODELLING

This part briefly reviews the concept of nonholonomic system and the definition of Lie bracket of vector fields, $X_1(q)$ and $X_2(q)$, in matrix form from the Cartesian (x, y) coordinate system:

$$\left[X_1, X_2\right](p) = \frac{\partial X_2}{\partial q} X_1\Big|_p - \frac{\partial X_1}{\partial q} X_2\Big|_p = \left[X_2(f)X_1 - X_1(f)X_2\right], \tag{1}$$

where $\frac{\partial X_1}{\partial q}$, $\frac{\partial X_2}{\partial q}$ are Jacobian matrices, X_1 and X_2 are vector fields on a smooth m-dimensional manifold M of $(q^1, q^2, ..., q^m)$ around some point $p \in M$ and $\left[X_1, X_2\right]$ is the Lie bracket. The nonlinear motions of the vehicle can be presented via this Lie bracket vector field.

Consider a vehicle rolling without slipping, the vehicle dynamics can be written in a set of first-order differential equations from its configuration variables. If the vehicle has the rear-wheel driving, the vehicle kinematic model, shown in figure 1, can be derived in equation (2):

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\varphi} \end{bmatrix} = \underbrace{\begin{bmatrix} \cos\theta \\ \sin\theta \\ \frac{\tan\varphi}{l} \\ 0 \end{bmatrix}}_{X_1} u_1 + \underbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}}_{X_2} u_2 \tag{2}$$

From (2), the four components of function X_1 are: $X_1^1 = \cos\theta$, $X_1^2 = \sin\theta$, $X_1^3 = \frac{\tan\varphi}{l}$, and $X_1^4 = 0$. The four components of function X_2 are: $X_2^1 = 0$, $X_2^2 = 0$, $X_2^3 = 0$, and $X_2^4 = 1$.

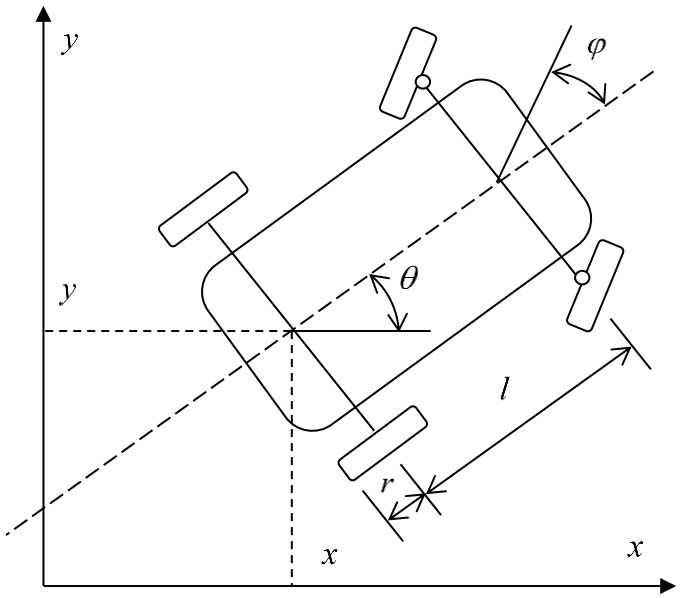

Figure 1. A simplified vehicle model

In figure 1, r is the vehicle wheel radius and l is the base length; x and y are the Cartesian coordinates of the rear wheel, θ measures the orientation of the vehicle body with respect to the x axis, and φ is the steering angle.

In equation (2), the vehicle motion is controlled by two inputs, u_1 is the linear driving velocity, and, u_2 is the angular steering velocity. There are four (4) state variables,

namely the position of the vehicle $x_1 = x$ and $x_2 = y$; the angle of the vehicle body orientation with respect to the x axis, $x_3 = \theta$; and the steering angle, $x_4 = \varphi$.

A useful tool to test the controllability of this nonlinear system in (2) is the use of Lie brackets rank: If the Lie brackets in (3) have full rank, the system is controllable.

$$rank\left[X_1, X_2, X_3, X_4\right] = \left[X_1, X_2, [X_1, X_2], [X_1, [X_1, X_2]]\right] \tag{3}$$

It can be seen that Jacobian matrix of the function X_1 is:

$$J_F(x, y, \theta, \varphi) = X_1(f) = \begin{bmatrix} \frac{\partial X_1^1}{\partial x} & \frac{\partial X_1^1}{\partial y} & \frac{\partial X_1^1}{\partial \theta} & \frac{\partial X_1^1}{\partial \varphi} \\ \frac{\partial X_1^2}{\partial x} & \frac{\partial X_1^2}{\partial y} & \frac{\partial X_1^2}{\partial \theta} & \frac{\partial X_1^2}{\partial \varphi} \\ \frac{\partial X_1^3}{\partial x} & \frac{\partial X_1^3}{\partial y} & \frac{\partial X_1^3}{\partial \theta} & \frac{\partial X_1^3}{\partial \varphi} \\ \frac{\partial X_1^4}{\partial x} & \frac{\partial X_1^4}{\partial y} & \frac{\partial X_1^4}{\partial \theta} & \frac{\partial X_1^4}{\partial \varphi} \end{bmatrix} = \begin{bmatrix} 0 & 0 & -\sin\theta & 0 \\ 0 & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & \frac{1}{l\cos^2\varphi} \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

And Jacobian matrix of the function X_2 is:

$$J_F(x, y, \theta, \varphi) = X_2(f) = \begin{bmatrix} \frac{\partial X_1^1}{\partial x} & \frac{\partial X_1^1}{\partial y} & \frac{\partial X_1^1}{\partial \theta} & \frac{\partial X_1^1}{\partial \varphi} \\ \frac{\partial X_2^2}{\partial x} & \frac{\partial X_2^2}{\partial y} & \frac{\partial X_2^2}{\partial \theta} & \frac{\partial X_2^2}{\partial \varphi} \\ \frac{\partial X_2^3}{\partial x} & \frac{\partial X_2^3}{\partial y} & \frac{\partial X_2^3}{\partial \theta} & \frac{\partial X_2^3}{\partial \varphi} \\ \frac{\partial X_2^4}{\partial x} & \frac{\partial X_2^4}{\partial y} & \frac{\partial X_2^4}{\partial \theta} & \frac{\partial X_2^4}{\partial \varphi} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

From equation (1), the Lie bracket of vector field $X_3 = \left[X_1, X_2\right]$ is:

$$X_3 = \left[X_1, X_2\right] = \left[X_2(f)X_1 - X_1(f)X_2\right]$$

$$= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \cos\theta \\ \sin\theta \\ \frac{\tan\varphi}{l} \\ 0 \end{bmatrix} - \begin{bmatrix} 0 & 0 & -\sin\theta & 0 \\ 0 & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & \frac{1}{l\cos^2\varphi} \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -\frac{1}{l\cos^2\varphi} \\ 0 \end{bmatrix} \tag{4}$$

And the Lie bracket of vector field $X_4 = \left[X_1\left[X_1, X_2\right]\right]$ is:

$$X_4 = \left[X_1\left[X_1, X_2\right]\right] =$$

$$= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\frac{2\tan\varphi}{l\cos^2\varphi} \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \cos\theta \\ \sin\theta \\ \frac{\tan\varphi}{l} \\ 0 \end{bmatrix} - \begin{bmatrix} 0 & 0 & -\sin\theta & 0 \\ 0 & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & \frac{1}{l\cos^2\varphi} \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ -\frac{1}{l\cos^2\varphi} \\ 0 \end{bmatrix} = \begin{bmatrix} -\frac{\sin\theta}{l\cos^2\varphi} \\ \frac{\cos\theta}{l\cos^2\varphi} \\ 0 \\ 0 \end{bmatrix} \tag{5}$$

Check the controllability of the 4x4 Jacobi-Lie bracket matrix in (3):

$$\det\left(X_1X_2X_3X_4\right)=\begin{pmatrix}\cos\theta & 0 & 0 & -\dfrac{\sin\theta}{l\cos^2\varphi}\\ \sin\theta & 0 & 0 & \dfrac{\cos\theta}{l\cos^2\varphi}\\ \dfrac{\tan\varphi}{l} & 0 & -\dfrac{1}{l\cos^2\varphi} & 0\\ 0 & 1 & 0 & 0\end{pmatrix}=\frac{1}{l^2\cos^4\varphi} \tag{6}$$

So, if $\varphi\neq\frac{\pi}{2}$, then $\det\left[X_1X_2X_3X_4\right]$ is well defined and the system in (2) is fully controllable. This means that the system in (2) can be transformed from any given state to any other state and all of its state parameters are under controlled by the two input vectors.

The vehicle model in (2) is nonlinear and has the first order derivative form:

$$\dot{X}=f(x,u) \tag{7}$$

where the state variables are $x\triangleq\left[x,y,\theta,\varphi\right]'$, and the inputs are $u=\left[u_1,u_2\right]'$. The nonlinear equation in (7) can be expanded in Taylor series around the reference setpoints (x_r,u_r) at $\dot{X}_r=f(x_r,u_r)$, that:

$$\dot{X}\approx f(x_r,u_r)+f_{x,r}(x-x_r)+f_{u,r}(u-u_r) \tag{8}$$

where $f_{x.r}$ and $f_{r.x}$ are the Jacobean of f corresponding to x and u, evaluated around the reference setpoints (x_r,u_r).

Subtraction of (8) and $\dot{X}_r=f(x_r,u_r)$ results a linear approximation to the system at the reference setpoints in continuous time (t):

$$\dot{\tilde{X}}(t)=A(t)\tilde{X}(t)+B(t)\tilde{u}(t) \tag{9}$$

where $\tilde{X}(t)=X(t)-X_r(t)=\begin{bmatrix}x(t)-x_r(t)\\ y(t)-y_r(t)\\ \theta(t)-\theta_r(t)\\ \varphi(t)-\varphi_r(t)\end{bmatrix}$, and $\tilde{u}(t)=u(t)-u_r(t)=\begin{bmatrix}u_1(t)-u_{r1}(t)\\ u_2(t)-u_{r2}(t)\end{bmatrix}$,

$$A(t)=\begin{bmatrix}0 & 0 & -u_{r1}(t)\sin\theta_r(t) & 0\\ 0 & 0 & u_{r1}(t)\cos\theta_r(t) & 0\\ 0 & 0 & 0 & \dfrac{u_{r1}(t)}{l\cos^2\varphi_r(t)}\\ 0 & 0 & 0 & 0\end{bmatrix},\ B(t)=\begin{bmatrix}\cos\theta_r(t) & 0\\ \sin\theta_r(t) & 0\\ \dfrac{\tan\varphi_r(t)}{l} & 0\\ 0 & 1\end{bmatrix}$$

It is noted that the linearized model in (9) is a time variant system depending on the current time (t).

The continuous system in (9) can be transformed to a discrete-time (k) with a scanning interval, $k+1=k+\Delta t$, and, Δt is the length of the sampling interval. The inputs $u(k)$

are held constant during the time interval $(k+1)$ and (k). The symbols of $x_k = x(k)$ and $u_k = u(k)$ are also used:

$$\begin{aligned} \tilde{X}(k+1) &= A(k)\tilde{X}(k) + B(k)\tilde{u}(k) \\ \tilde{Y}(k) &= C(k)\tilde{X}(k) \end{aligned} \tag{10}$$

where,

$$A(k) = \begin{bmatrix} 1 & 0 & -u_{r1}(k)\sin\theta_r(k)(\Delta t) & 0 \\ 0 & 1 & u_{r1}(k)\cos\theta_r(k)(\Delta t) & 0 \\ 0 & 0 & 1 & \dfrac{u_{r1}(k)}{l\cos^2\varphi_r(k)}(\Delta t) \\ 0 & 0 & 0 & 1 \end{bmatrix},\ B(k) = \begin{bmatrix} \cos\theta_r(k)(\Delta t) & 0 \\ \sin\theta_r(k)(\Delta t) & 0 \\ \dfrac{\tan\varphi_r(k)}{l}(\Delta t) & 0 \\ 0 & (\Delta t) \end{bmatrix},$$

$$C(k) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \text{ and,}$$

$$\tilde{X}(k) = X(k) - X_r(k) = \begin{bmatrix} x(k) - x_r(k) \\ y(k) - y_r(k) \\ \theta(k) - \theta_r(k) \\ \varphi(k) - \varphi_r(k) \end{bmatrix},\ \tilde{u}(k) = u(k) - u_r(k) = \begin{bmatrix} u_1(k) - u_{r1}(k) \\ u_2(k) - u_{r2}(k) \end{bmatrix}.$$

In this discretized model, the two control inputs are the difference in the actual and the desired velocity, $u_1(k) - u_{r1}(k)$, and, $u_2(k) - u_{r2}(k)$. The four outputs, $y(k) = \tilde{Y}(k) = C(k)\tilde{X}(k)$, are assumed totally to be measured and updated in real-time scanning interval from the GPS map. It is noted that the vehicle discretized model in (10) is a time variant system and its transfer function is depending on its positions and the scanning speeds. Equations (10) are used to develop MPC algorithms in the next part.

3. MODEL PREDICTIVE CONTROL

This part presents the design of MPC algorithms for the optimal open-loop optimization problem that minimizes the difference between the predicted vehicle behavior and the desired vehicle behavior. MPC differs from other control techniques in that the optimal control problem is solved on-line for the current state, rather than off-line determined as the feedback policy. MPC has been widely applied in the robotic and automation technologies because of its ability to handle the input and output constrains in the optimal control problem.

MPC algorithms are now developed to control the two (2) inputs of the vehicle driving velocity, $u_1(k)$, and, the vehicle steering velocity, $u_2(k)$, in order to achieve the four (4) desired outputs of the vehicle coordinate positions, $x_1(k) = x(k)$, and $x_2(k) = y(k)$; the vehicle orientation body angle with respect to the x axis, $x_3(k) = \theta(k)$; and the steering angle, $x_4(k) = \varphi(k)$.

From (10), the prediction horizon for the outputs, $y_{k+i|k}$, and the input increments, $\Delta u_{k+i|k}$, can be formulated as,

$$\begin{bmatrix} y_{k+1|k} \\ y_{k+2|k} \\ \vdots \\ y_{k+N_y|k} \end{bmatrix} = \begin{bmatrix} CA \\ CA^2 \\ \vdots \\ CA^{N_y} \end{bmatrix} x_{k|k} + \begin{bmatrix} CB \\ CAB+CB \\ \vdots \\ \sum_{i=1}^{N_y} CA^{i-1}B \end{bmatrix} u_{k-1} + \begin{bmatrix} CB & 0 & \cdots & 0 \\ CAB+CB & CB & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{i=1}^{N_y} CA^{i-1}B & \sum_{i=1}^{N_y-1} CA^{i-1}B & \cdots & \sum_{i=1}^{N_y-N_u+1} CA^{i-1}B \end{bmatrix} \begin{bmatrix} \Delta u_k \\ \Delta u_{k+1} \\ \vdots \\ \Delta u_{k+N_u-1} \end{bmatrix} \quad (11)$$

In MPC, we seek to minimize a cost function that based on the deviation of the states and the inputs from their targets to guarantee stability and to achieve the best possible performance. Therefore, we seek the input sequence $U \triangleq \left\{\Delta u_k,...,\Delta u_{k+N_u-1}\right\}$ that minimize a cost function $\min_{U\triangleq\left\{\Delta u_k,...,\Delta u_{k+N_u-1}\right\}} J(U,x(k))$. In general, this cost function is defined as:

$$\min_{U\triangleq\left\{\Delta u_k,...,\Delta u_{k+N_u-1}\right\}} \left\{ \begin{array}{l} J(U,x(k)) = x_{k+N_y}^{'} P_{LYA} x_{k+N_y} + \\ \sum_{i=0}^{N_y-1} \left[(y_{k+i|k} - r_{k+i|k})^{'} Q (y_{k+i|k} - r_{k+i|k}) + \Delta u_{k+i|k}^{'} R \Delta u_{k+i|k} \right] \end{array} \right\} \quad (12)$$

where x_k denotes the state variables at the current discrete time (*k*): P_{LYA} is the calculated Lyapunov matrix to assure the stability of this time variant system; $U \triangleq \left\{\Delta u_k,...,\Delta u_{k+N_u-1}\right\}$ is the solution of input increments, N_u is the inputs predictive horizon; N_y is the outputs predictive horizon; $y_{k+i|k}$ are the predictive outputs at the current discrete time (*k*), $r_{k+i|k}$ are the corresponding reference output setpoints; $\Delta u_{k+i|k}$ are the input increments prediction with $\Delta u_{k+i|k} = u_{k+i|k} - u_{k+i-1|k}$; $Q = Q^{'} \geq 0$, $R = R^{'} > 0$ are the weighting penalty matrices for predicted outputs and input increments, respectively.

The simplest possibility to enforce the stability with a finite prediction horizon is to add a so-called zero-terminal equality at the end of the prediction horizon as per Minh V.T and Afzulpurkar N. (2006) [5]. Therefore we force all states $x_{k+N_y} = 0$ after N_y prediction horizon and then, the Lyapunov term in (12) $x_{k+N_y}^{'} P_{LYA} x_{k+N_y} = 0$ (zero terminal).

The zero terminal equality constraint poses difficulty to the optimization algorithm since all the system states have to merge to zero after a finite horizon prediction length. Therefore, choosing this horizon length may be challenging because if, N_y, is not chosen to be long enough, the objective function in (12) may not have a feasible solution. Another shortfall of the zero terminal equality constraint is that the solution to the minimization problem often takes roughen (non-smooth) path in order to satisfy the constraint. However in the size of this paper, we will test only for zero terminal equality MPC. Other methods such as terminal regions will be investigated in the next step of this research.

From (12), when put the term $x_{k+N_y}^{'} P_{LYA} x_{k+N_y} = 0$, a tracking setpoints MPC objective function with hard constraints can be developed:

$$\min_{U\triangleq\left\{\Delta u_k,...,\Delta u_{k+N_u-1}\right\}} \left\{ J(U,x(k)) = \sum_{i=0}^{N_y-1} \left[(y_{k+i|k} - r_{k+i|k})^{'} Q (y_{k+i|k} - r_{k+i|k}) + \Delta u_{k+i|k}^{'} R \Delta u_{k+i|k} \right] \right\}$$

, (13)

subject to:

$u_k \in \mathcal{U}$, and $u_{k+i} \in [u_{\min}, u_{\max}]$, $\Delta u_{k+i} \in [\Delta u_{\min}, \Delta u_{\max}]$, for $i = 0,1,...,N_u - 1$,

$y_k \in \mathcal{Y}$, and $y_{k+i|k} \in [y_{\min}, y_{\max}]$, for $i = 0,1,...,N_y - 1$,

$\Delta u_k = u_k - u_{k-1} \in \Delta\mathcal{U}$, and $\Delta u_{k+i} = 0$, for $i \geq N_u$,
$x_{k|k} = x(k)$, $x_{k+i+1|k} = A(k)x_{k+i|k} + B(k)u_{k+i}$, $u_{k+i|k} = u_{k+i-1|k} + \Delta u_{k+i|k}$, $y_{k+i|k} = C(k)x_{k+i|k}$.

The MPC regulator computes the optimal solution, $U^* \triangleq \left\{\Delta u_k{}^*,...,\Delta u_{k+N_u-1}{}^*\right\}$ and generates the new inputs $u_{k+i|k} = u_{k+i-1|k} + \Delta u_{k+i|k}$, from the objective function (13), then applies only the first element of the current inputs increment, $\Delta u_k{}^*$, to the current optimal inputs, $u^*(k) = u_{k-1} + \Delta u_k{}^*$. After having inserted the current optimal inputs, the MPC regulator repeats the optimization, $u^*(k+1)$, for the next interval time, $k+1$, based on the new update state variables $x(k+1)$. This way, the closed loop control strategy is obtained by solving on-line the open loop optimization problem.

By substituting $x_{k+N_y|k} = A^{N_y}(k)x(k) + \sum_{i=0}^{N_y-1} A^i(k)B(k)u_{k+N_y-1-i}$, equation (13) can be rewritten as a function of only the current state $x(k)$ and the current setpoints $r(k)$:

$$\Psi\left(x(k), r(k)\right) = \frac{1}{2}x'(k)Yx(k) + \min_U \left\{\frac{1}{2}U'HU + x'(k)r(k)FU\right\}, \tag{14}$$

subject to the hard combined inputs/outputs constraints of $GU \leq W + Ex(k)$, where the column vector $U \triangleq \left[\Delta u_k,...,\Delta u_{k+N_p-1}\right]' \in \Delta\mathcal{U}$ is the prediction optimization vector; $H = H' > 0$, and H, F, Y G, W and E are matrices obtained from Q, R and given constraints in (13). As only the optimizer U is needed, the term involving Y is usually removed from (14). Then, the optimization problem in (14) is a quadratic program and depends only on the current state $x(k)$ and the current setpoints $r(k)$ subject to the hard combined constraints. The implementation of MPC requires an on-line solution of this quadratic program at each time interval (k).

In reality, the system would have both input and output constraints and the difficulty will arise due to the inability to satisfy the output constraints due to the constraints in inputs. Since MPC is designed for on-line implementation, any infeasible solution of the online optimization problem in (14) cannot be allowed. Normally the input constraints are based on the physical limits of the vehicle. If the outputs constraints are on tracking position errors, they are not very strictly imposed and can be violated somewhat during the evolution of the performance. To guarantee the system stability once the outputs violate the constraints, the hard constrained optimization in (13) can be modified to a new MPC objective function with some softened constraints as:

$$\min_{U \triangleq \left\{\Delta u_k,...,\Delta u_{k+N_u-1}\right\}} \left\{J(U, x(k)) = \sum_{i=0}^{N_y-1}\left[(y_{k+i|k} - r_{k+i|k})'Q(y_{k+i|k} - r_{k+i|k}) + \Delta u'_{k+i|k}R\Delta u_{k+i|k} + \varepsilon_i'(k)\Lambda\varepsilon_i(k)\right]\right\} \tag{15}$$

,

where $\varepsilon_i(k) \geq 0$ are the new penalty terms added to the MPC objective function, $\varepsilon_i(k) = \left[\varepsilon_y; \varepsilon_u\right]$, $y_{\min} - \varepsilon_y \leq y_{k+i|k} \leq y_{\max} + \varepsilon_y$ and $u_{\min} - \varepsilon_u \leq u_{k+i|k} \leq u_{\max} + \varepsilon_u$. And $\Lambda = \Lambda' \geq 0$ is the new penalty matrix (usually $\Lambda > 0$ and set with small values). These terms, $\varepsilon_i(k)$, will keep the constrained violations at low values until the solution is returned. A new MPC algorithm for softened constraints to select the optimal inputs $u^*(k+i\,|\,k)$ can be conducted similarly to (14) with the new added penalty terms $\varepsilon_i'(k)\Lambda\varepsilon_i(k)$.

Furthermore, in order to increase the possibility of the MPC to find out solution in some critical time, some output setpoints can be temporally deleted because the deletion of some output setpoints can make the system looser and the probability that the MPC optimizer can find a solution will increase. Deletion of some output setpoints can be

implemented via temporally assigning some zeros in the penalty matrices Q and R. For example, the above MPC controller has four outputs $y=\left[y_1, y_2, y_3, y_4\right]^{'}$, if we select the 4 by 4 penalty matrix $Q=diag\{1,1,1,1\}$, implying that all four outputs are required to reach their setpoitnts. But if we want to delete the output setpoints for y_3, y_4 or it is required that only the two outputs, y_1, y_2, to reach the setpoints, we can select a new penalty matrix of $Q=diag\{1,1,0,0\}$. In this case, the new controlled variables now become $y=\left[y_1, y_2\right]^{'}$ as the penalties for y_3, y_4 have been deleted.

The robustness of MPC can be also increased if some setpoints can be relaxed into regions rather than put in some specific values. Then, another new MPC algorithm can be developed if the setpoints $r(k)$ can be changed into regions. An output region is defined by the minimum and maximum values of a desired range. The minimum value is the lower limit, and the maximum value is the upper limit and satisfied $y_{lower} \le y_{k+i|k} \le y_{upper}$. The modified objective function for this MPC with output regions is:

$$\min_{U \triangleq \left\{\Delta u_k, \ldots, \Delta u_{k+N_u-1}\right\}} \left\{ J(U, x(k)) = \sum_{i=0}^{N_y-1} \left[z_{k+i|k}^{'} Q z_{k+i|k} + \Delta u_{k+i|k}^{'} R \Delta u_{k+i|k} \right] \right\}, \tag{16}$$

where $z_{k+i|k} \ge 0$; $z_{k+i|k} = y_{k+i|k} - y_{upper}$ for $y_{k+i|k} > y_{upper}$;

$z_{k+i|k} = y_{lower} - y_{k+i|k}$ for $y_{k+i|k} < y_{lower}$; $z_{k+i|k} = 0$ for $y_{lower} \le y_{k+i|k} \le y_{upper}$

As long as the outputs in (16) still lie inside the desired regions, no control actions are taken because none of the control objectives have been violated, all $z_{k+i|k} = 0$ (in this case, we imagine that the vehicle has been tracked well on its stable trajectory and within a desired region). But when an output violates the desired region, the MPC regulator will be activated and push them back to the desired regions. This modified MPC objective function can help to make the vehicle tracking smoother and the controller tasks can be reduced.

Simulations for the MPC application are presented in the following next part.

4. MPC FOR TRACKING SETPOINTS

For the trajectory tracking, a reference trajectory is generated by solving the vehicle differential equations in (2). The difference of the reference trajectory setpoints and the actual current vehicle positions is provided at the real time to the MPC regulator. The MPC regulator calculates the optimized control inputs and only the first element of this optimal solution is fed into system to generate the next outputs. The updated outputs are now compared with the updated setpoints for the next MPC regulator calculation. The diagram of the MPC control system is shown in figure 2.

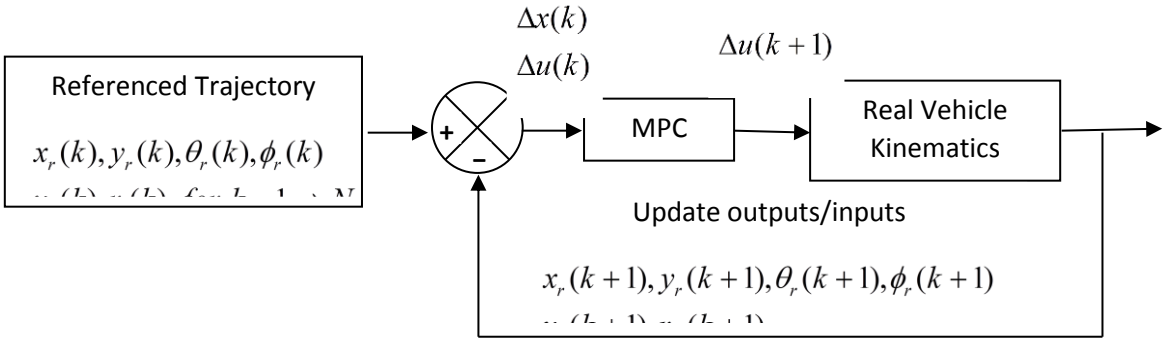

Figure 2. MPC control system

We implement this MPC system to track a full circle trajectory. For generating online this full circle trajectory, the reference desired inputs are set at $u_1 = r\omega$ and $u_2 = 0$. The initial reference positions are set at $\begin{bmatrix} x_{r0} & y_{r0} & \theta_{r0} & \varphi_{r0} \end{bmatrix}' = \begin{bmatrix} 0 & 0 & 0 & \arctan\dfrac{r}{l} \end{bmatrix}'$. For this simulation, we set the vehicle parameters at $r = 0.5m$, $l = 1.5m$, and $\omega = 1rad/\sec$. The reference setpoints are generated using ODE45 in MATLAB and shown in figure 3.

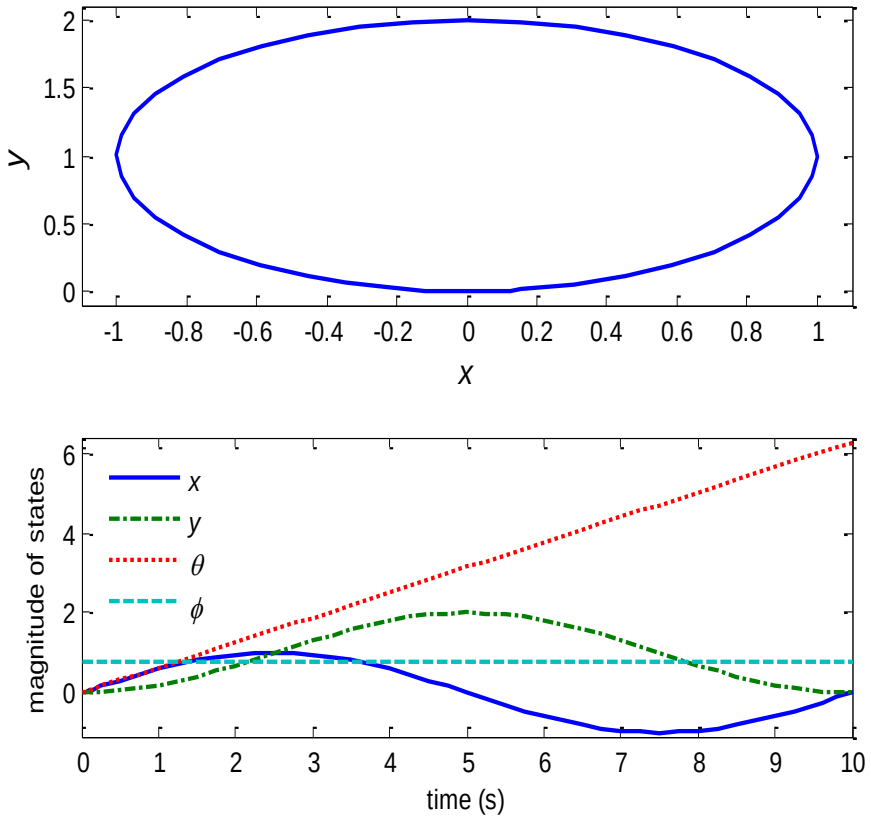

Figure 3. Circular reference setpoints

We now use the MPC control system in figure 2 to track the online reference setpoints of $x_r(k)$, $y_r(k)$, $\theta_r(k)$, and $\varphi_r(k)$ in figure 3.

For this simulation, the initial positions of the vehicle are set at $X_0 = \begin{bmatrix} -0.5 & -0.5 & 0 & 0 \end{bmatrix}'$; The constraints are set at $u_{\min} = \begin{bmatrix} -1, -1 \end{bmatrix}'$, $u_{\max} = \begin{bmatrix} 1,1 \end{bmatrix}'$, $\Delta u_{\min} = \begin{bmatrix} -0.5, -0.5 \end{bmatrix}'$, $\Delta u_{\max} = \begin{bmatrix} 0.5, 0.5 \end{bmatrix}'$, $y_{\min} = \begin{bmatrix} -1, -1, -1, -1 \end{bmatrix}'$, and $y_{\max} = \begin{bmatrix} 1,1,1,1 \end{bmatrix}'$; The predictive horizons are set equally at $N_u = 10$ and $N_y = 10$; The penalty matrices are set at $Q = diag\{1,1,1,1\}$ and $R = diag\{1,1\}$. Performance of this MPC to track the circular reference is shown in figure 4. The MPC optimizer is minimizing the tracking errors $y_{k+i|k} - r_{k+i|k}$ at each intervals during its evolution performance from the initial position, $error_{initial} = y_{k|0} - r_{k|0} = \begin{bmatrix} -0.5 \\ -0.5 \\ 0 \\ 0 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0.7854 \end{bmatrix} = \begin{bmatrix} -0.5 \\ -0.5 \\ 0 \\ -0.7854 \end{bmatrix}$ to the final position,

$error_{final} = y_{k|N} - r_{k|N} = \begin{bmatrix} -0.0202 \\ 0.0257 \\ 0.0094 \\ 0.7648 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0.7854 \end{bmatrix} = \begin{bmatrix} -0.0202 \\ 0.0257 \\ 0.0094 \\ -0.0206 \end{bmatrix}$, or very small errors are left at the end of the tracking trajectory.

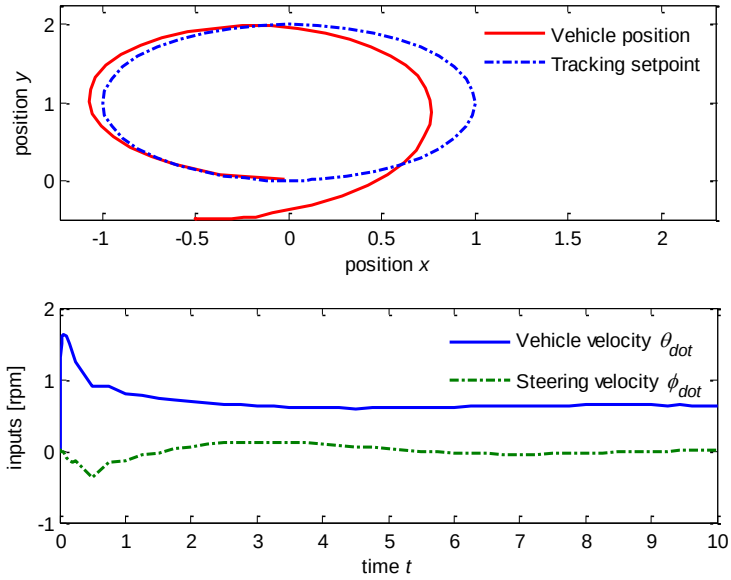

Figure 4. Tracking MPC linearized model

If the MPC prediction horizon is shortened, the online calculation burden will be considerably reduced but it will lead to faster incremental changes of the inputs and then, bad performance of the outputs. With shortened horizon predictions, the controller may become instable. Figure 5 shows the MPC performance with shortened predictive horizons to $N_u = 4$ and $N_y = 4$.

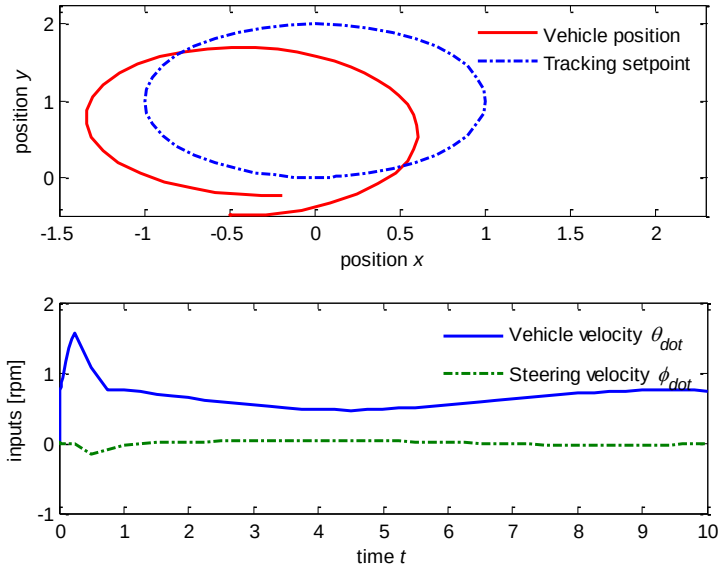

Figure 5. MPC linearized model with short horizon

Sufficient long prediction horizon will increase the MPC performance and its stability. However, the calculation burden of this system will be dramatically increased. The next MPC simulation runs with $N_u = 20$ and $N_y = 20$ shown in figure 6. Performance of these tracking outputs is much improved as well as the inputs become smoother or easier to be regulated.

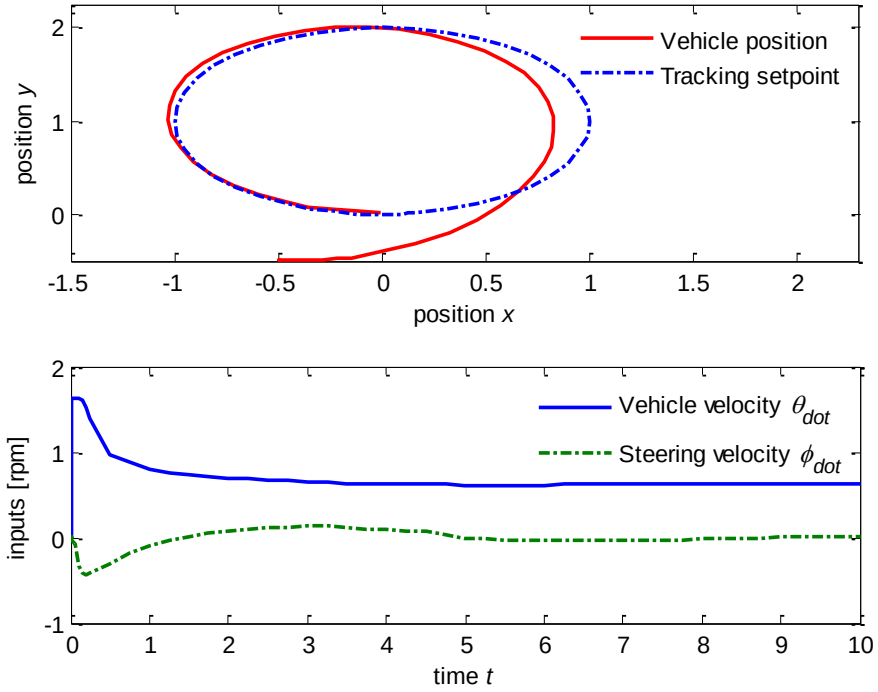

Figure 6. MPC linearized model with long horizon

However with too long horizon length, MPC will result too slow control increments and therefore deteriorate the outputs performance. The MPC system becomes instable as shown in figure 7 with too long prediction horizon of $N_u = 23$ and $N_y = 23$.

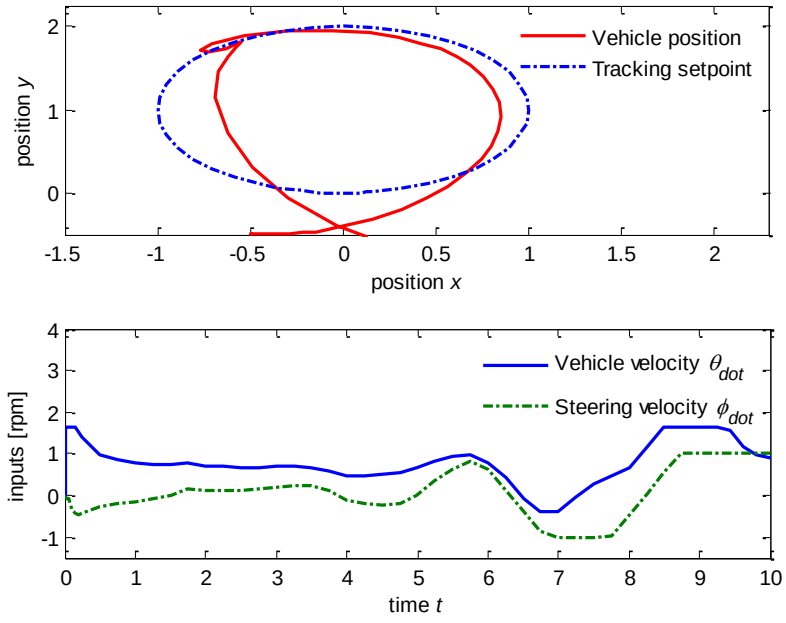

Figure 7. MPC linearized model with too long horizon and instability

Regulation of the penalty matrices can also help to change the MPC performance. If we set $Q \gg R$ (Q is set much bigger than R), then any small changes of the outputs will affect dramatically to the MPC objective function. It means that the inputs are set to be changed faster and easier than the outputs. But, the vehicle inputs (speeds) are harder to be regulated or changed, so we must sacrifice some tracking output errors to gain some smoother inputs by setting $Q \ll R$. The next simulation shown in figure 8 runs with $N_u = 6$, $N_y = 6$, $Q = diag\{1,1,1,1\}$, and $R = diag\{10,10\}$. Figure 7 shows that the inputs become smoother but the outputs tracking errors become considerably larger.

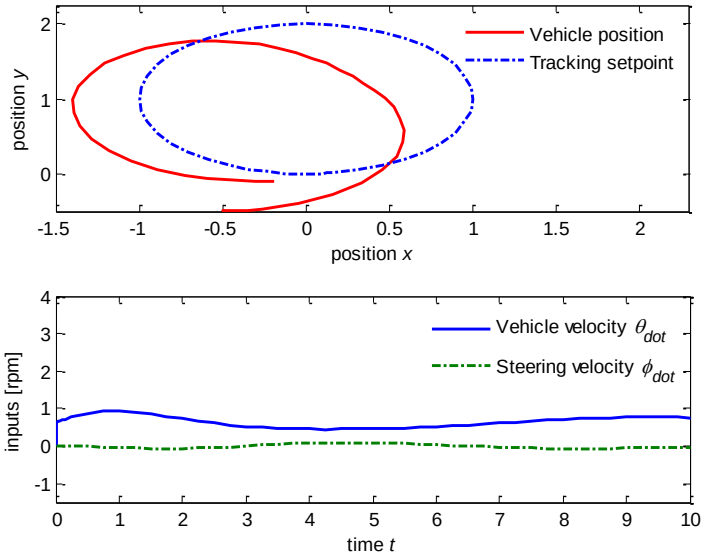

Figure 8. MPC linearized model with $Q \ll R$

Inversely, we set now $N_u = 6$, $N_y = 6$, $Q = diag\{10,10,10,10\}$, and $R = diag\{1,1\}$. Figure 9 shows that the system becomes very sensitive to the input changes. These faster input changes can be seen in the triangular shape. These inputs shape is unrealistic or we cannot control the vehicle velocity on that shape. And consequently, the system becomes instability.

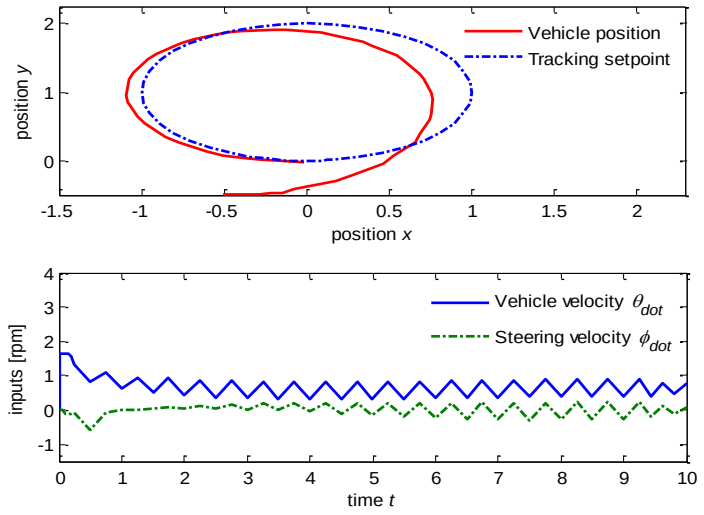

Figure 9. MPC linearized model with $Q \gg R$

Another way to regulate the controller is to change the reference setpoint errors ($y_k - y_r$). In the previous simulations, we set these errors to zeros, $r_k = y_k - y_r = [0,0,0,0]'$. But in order to offset the vehicle sideslips or to compensate some possible model-plant mismatches, we can change some setpoint errors. For example, if we set $r_k = [0.1,0.1,0,0]'$, the MPC performance is shown in figure 10, the final position of the vehicle becomes $[x_F, y_F] = [0.1,0.1]$, but the vehicle tracks faster to the reference trajectory.

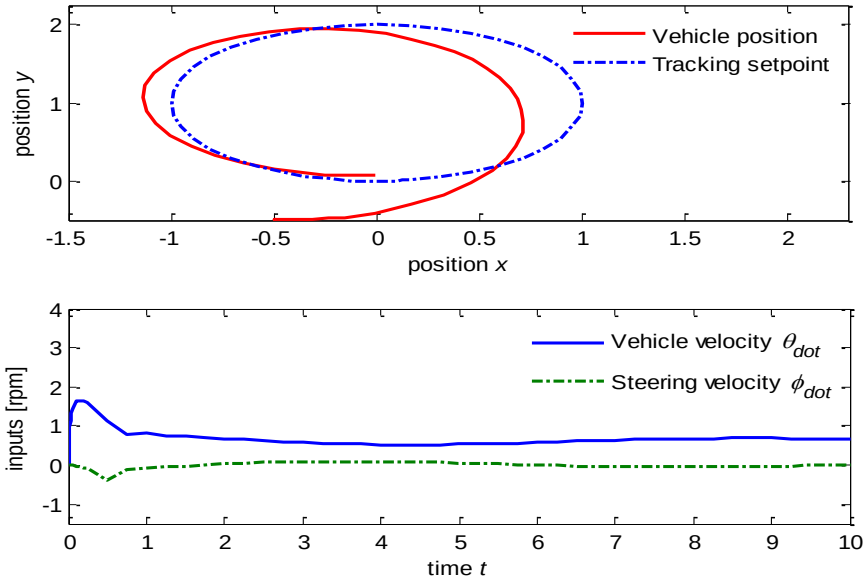

Figure 10. MPC linearized model with setpoint offsets in positions

We can also regulate the error offsets for the vehicle orientation angle, θ, and the steering angle, φ. For example, if we set the tracking errors at $r_k = [0,0,0.1,0.1]'$, the MPC performance is shown in figure 11. Due to the positive error offsets on the orientation and the steering angle, the vehicle rotates in a smaller radius and arrives the destination at $[\theta_F, \varphi_F] = [0.1,0.1]$.

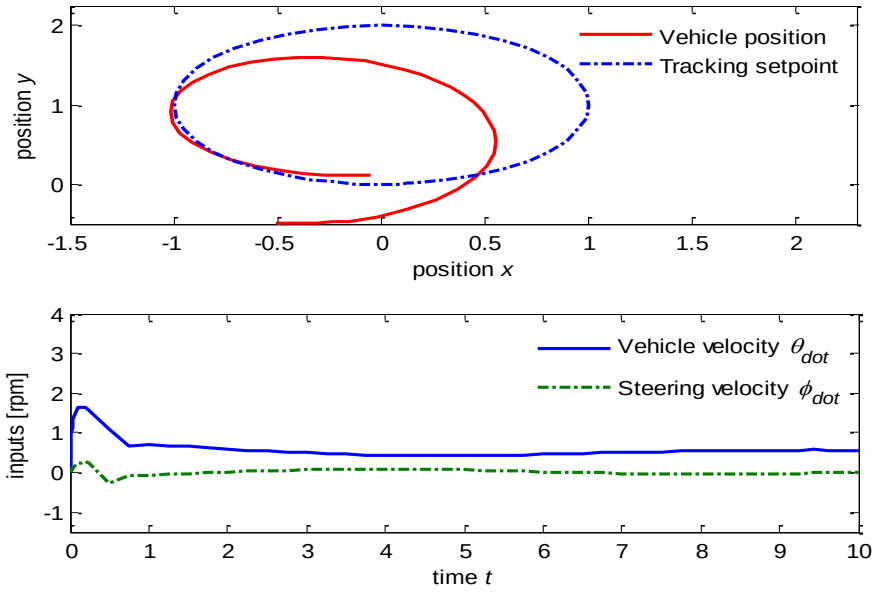

Figure 11. MPC linearized model with setpoint offsets in angles

In this paper, all MPC performances are set for the vehicle initial position at $X_0^{\text{Vehicle}} = \begin{bmatrix} -0.5 & -0.5 & 0 & 0 \end{bmatrix}'$. It is a big difference to the initial position of the reference trajectory of $X_0^{\text{Reference}} = \begin{bmatrix} 0 & 0 & 0 & \arctan\dfrac{r}{l} \end{bmatrix}'$. This difference can be considered as some measured output errors or the model-plant mismatches. The MPC regulator can gradually eliminate these errors during its evolution and drive the vehicle closer to the reference setpoints.

5. CONCLUSION

MPC schemes for tracking setpoints have been developed and tested for controlling the vehicle tracking in a full referenced circle. Simulations show that MPC can control very well the tracking setpoints subject to constraints. The MPC performance, stabilization as well as the robustness can be regulated and improved by variant the MPC parameters as well as modifying its objective functions to softened constraints or to output regions. MPC schemes are able to guarantee the system stability even when the initial conditions lead to violations of some constraints.

Even though simulations show that modified MPC algorithms are successful in controlling the vehicle tracking, model of uncertainty and the model-plant mismatches that may affect the closed loop stability are still open issues. Further analysis is needed for the effectiveness of the modified MPC schemes to softened constraints and to output regions. Real experiments and other validations for this proposed technique are also needed in the next step of the project.

ACKNOWLEDGMENT

The author would like to thank Papua New Guinea Institute of Technology (UNITECH) for supporting this research project.

CONFLICT OF INTERESTS

The author declares that there is no conflict of interest regarding the publication of this research article.

REFERENCES

[1] Minh V.T and Hashim F.B. "Tracking setpoint robust model predictive control for input saturated and softened state constraints".International Journal of Control, Automation and Systems, 2011; vol 9(5): 958–965. DOI: 10.1007/s12555-011-0517-4.

[2] Minh V.T, Hashim FB and Awang M. "Development of a Real-time Clutch Transition Strategy for a Parallel Hybridelectric Vehicle". Proc IMechE, Part I: J Systems and Control Engineering, 2012; vol 226(12): 188–203. DOI: 10.1177/0959651811414760.

[3] Minh V.T., Advanced Vehicle Dynamics, 1st edition, Malaya Press, Pantai Valley, 50603. Kuala Lumpur, Malaysia, 2012, pp. 127-144, ISBN: 978-983-100-544-6.

[4] Minh V.T and Afzulpurkar N., "Robust Model Predictive Control for Input Saturated and Softened State Constraints", Asian Journal of Control, 2005,vol 7(3) pp. 323-329, DOI: 10.1111/j.1934-6093.2005.tb00241.x.

[5] Minh V.T and Afzulpurkar N., "A Comparative Study on Computational Schemes for Nonlinear Model Predictive Control", Asian Journal of Control, 2006, vol 8(4) pp. 324-331, DOI: 10.1111/j.1934-6093.2006.tb00284.x.

[6] Falcone P., Borrelli F., Tseng H., Asgari J., and Hrovat D., "*A Hierarchical Model Predictive Control Framework for Autonomous Ground Vehicles*", American Control Conference (ACC), Seattle, Washington, USA, pp. 3719-3724, June 2008. ISSN : 0743-1619.

[7] Lei L, Zhurong J., tingting C., and Xinchung J., "Optimal Model Predictive Control for Path Tracking of Autonomous Vehicle", 3rd International Conference on Measuring Technology and Mechatronics Automation (ICMTMA), Shanghai, China, Vol. 2, pp. 791-794, February 2011. DOI: 10.1109/ICMTMA.2011.481.

[8] Bahadorian M, Savkovic B., Eston R, and Hesketh T. "Toward a Robust Model Predictive Controller Applied to Mobile Vehicle Trajectory Tracking Control", Proceedings of the 18th IFAC World Congress, Milano, Italia, vol. 18(1), pp. 552-557, September 2011, DOI: 10.3182/20110828-6-IT-1002.01786.

[9] Wang J, Lu Z, Chen W, and Wu X, "An Adaptive Trajectory Tracking Control of Wheeled Mobile Robots", Industrial Electronics and Applictions (ICIEA), Beijing, China, pp. 1156-1160, June 2011, DOI: 10.1109/ICIEA.2011.5975761.

[10] Shim T, Adireddy G, and Yuan H, "Autonomous Vehicle Collision Avoidance System using Path Planning and Model Predictive Control Base Active Front Steering and Wheel Torque Control", Part D: Journal of Automobile Engineering, 2012, vol. 226(6), pp. 767-778, DOI: 10.1177/0954407011430275.

[11] Minh V.T., "Clutch control and vibration reduction for a hybrid electric vehicle", Proc IMechE, Part I: J Systems and Control Engineering, vol 226(7), pp. 867-874, 2012. DOI: 10.1177/0959651812445842.

[12] Minh V.T., Trajectory Generation for Autonomous Vehicle, Mechatronics 2013, Springer, ISBN 978-3-319-02293-2, pp. 615-626, 2014. DOI: 10.1007/978-3-319-02294-9_78.