Sparse Predictive Control for Vehicle Motion

John Mark Vergara Batabat
Department of Electrical and Computer Engineering
Instituto Superior Técnico - Universidade de Lisboa, Portugal
Email: john.batabat@tecnico.ulisboa.pt

***Abstract*—This work proposes the development of a control strategy based on the use of ℓ_1-norm penalty to the cost of Model Predictive Control (MPC) scheme for autonomous vehicles. The resulting control strategy will then be called sparse predictive control or sparse controller, because of the type of action considered. The vehicle is modeled using a discretized version of single integrator model expressed in Cartesian coordinates for a linear formulation, and a discretized version of unicycle model expressed in polar coordinates for a nonlinear formulation. Simulation results are provided to validate and evaluate the viability of the proposed control strategy in relation to the optimization parameters and constraints as applied in a known environment inhabited with given obstacles. A comparison analysis to quadratic or conventional cost function of MPC is also conducted. Based on the results from simulations, the feasibility of the proposed control strategy for vehicle motion control is shown as well as the effect of the selection of the cost weights that configure the controller. Furthermore, it can suggest that sparse controller is a better alternative to a conventional implementation of MPC, especially on the basis of minimizing the cost incurred during the process, resulting to a minimal fuel consumption.**

***Index Terms*—Sparsity, Model Predictive Control, Receding Horizon Control, Mobile Robot, Obstacle Avoidance.**

I. Introduction

There is a rapid increase of interest on Autonomous Vehicles (AV), that includes airborne, space borne, ground, and submersible. In an AV, it is appropriate to ask what will be the way to control a vehicle for a given obstacle avoidance situation. This assessment can be done in the context of Model Predictive Control (MPC), one of the modern control methods. The development of these modern control concepts can be traced back to the works of Kalman in the early 1960's [1], [2]. Kalman formulated in his two ground-breaking papers a precise design for the optimal least squares transient control problem and later on became to be known as Linear Quadratic Regulator (LQR) and Linear Quadratic Gaussian (LQG). These works became a standard approach to solve control problems in a wide range of application areas [3]. But it made a less significant impact on control technology development in the process industries. These failures can be traced to:

1) its incapacity to deal with constraints
2) its incapacity to solve process nonlinearities
3) its uncertainty in modeling
4) uniqueness of its cost function
5) cultural reasons.

Because of the these shortcomings, it led to the development of a more advanced control method. In the late 1970's, various papers were released showing applications of MPC in the industry. Two of the most well-known papers were from the works of Richalet et al. [4] presenting MPC as Model Algorithmic Control (MAC) and of the Shell Oil company engineers Cutler and Ramakter [5] for their work on Dynamic Matrix Control (DMC) initially intended for its use in petroleum refineries. However, the earliest patent appears to be granted to Martin-Sanchez in 1976, who called his method simply Adaptive Predictive Control [6]. An adaptive MPC technique approach developed by Clarke et al., Generalized Predictive Control (GPC) [7], [8], has also received considerable attention. The term "predictive" appears in the name of GPC because of the application of minimum variance in predicted values on the finite future time. Additionally, the term "predictive" also appears in MPC since the objective function is given in predictive values on the finite future time that can be computed by using the model [9].

In recent years the MPC landscape has changed drastically. Although some other options are considered, such as the so called Economic MPC [18], currently, the typical MPC formulations are commonly structured with quadratic cost function, both in the input and output cost. The choice is mainly due to its simplicity and to the possibility of making use of the properties from the LQR. Even though MPC have already been implemented in many ways and widely studied for AV, it is also of great interest to develop such controllers that will guarantee desirable path planning while considering fuel efficiency or minimality in the control effort. This proposition could be achieved by implementing the sparsity on its cost function. The use of ℓ_1-norm regularization in the sparsity formulation is typically known to force many elements in the control set to be equal to zero. As a result, it takes a little effort in control(that is, more fuel-efficient for vehicles) by a substantial margin compare to the other regularization terms.

In this dissertation, sparsity is incorporated with MPC to develop a control strategy which will be later called the sparse predictive control or sparse controller. Sparsity has become a significant topic of researchers in various fields. It has been successfully applied to some real-world applications, mostly in the fields of image processing and computer vision [10]. In the field of controls, modifications of conventional quadratic MPC methods have been developed to obtain spatially or temporally sparse control signals. One paper presented two stabilizing ℓ_1-regularized MPC controllers for redundantly-actuated Linear Time-Invariant (LTI) systems, by means of terminal constraint set and terminal cost[11]. However, the

paper only focuses on the investigation of theoretical basis in enhancing optimality, enlarging the domain of attraction, and minimizing conservativeness of the candidate terminal controllers. Another literature examines the effectiveness of sum-of-norms regularized with a numerical example for controlling a drone [12]. Though the paper gives a good modification of the conventional cost function of MPC, it emphasizes only the use of the euclidean norm to sparsify the solution.

The objective of this thesis is to develop a control strategy and study the implementation of the ℓ_1-norm, as well as the ℓ_2-norm in the regularization term or the control input cost. Then, a comparison of the performance of both implementations will be performed, enough to explain and deduce a conclusion on the sparsity idea. Both controllers are also simulated in an environment containing obstacles with different shapes to examine the extent of their capacities to handle hard constraints.

II. Model Predictive Control Background

MPC is a form of control scheme based on iterative, finite-horizon optimization problem to obtain for the current control action using the current state of the plant as the initial state [13]. This current state is used to make predictions of future values of the states or outputs. Then the optimization yields a sequence of optimal control moves (that is, manipulated input changes) so that the predicted response moves to the set point in an optimal manner [14]. In other words, it is used to predict a sequence of input control changes so that the output tries to follow the target or minimize the difference between them. Shown in Fig. (1) is the basic principle of MPC. According to a receding horizon strategy of this sequence of control values, only the first is actually applied to the plant and the process is repeated again obtaining a different optimal control sequence.

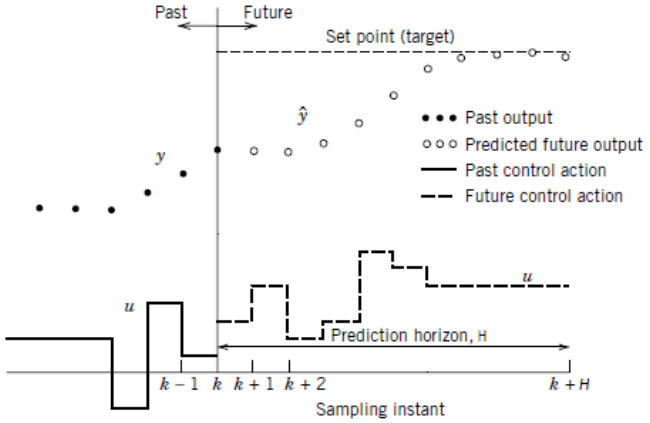

Figure 1. Model Predictive Control Basics (adapted from [14])

All MPC algorithms have the same basic elements or requirements for its implementation. They will only vary on the chosen options of the designers to modify one of these elements giving rise to different algorithms. These requirements include but not limited to the following:

1. Mathematical model of the plant
2. Cost function
3. Constraints
4. Prediction Model
5. Obtaining the control law

Further details are provided in the following sections for every requirement.

III. Linear Sparse Predictive Control

In most cases the actual mathematical model of a system is represented as an approximation to a linear model. Consider a linear plant model in its discrete-time state-space representation

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) \\ y(k) &= Cx(k) \end{aligned} \tag{1}$$

The representation from eq. (1) can be extended accordingly with a given prediction horizon, H. The result will be called as the prediction model with a generalized form for the state variable, x:

$$\begin{aligned} x(k+1|k) &= Ax(k|k) + Bu(k|k) \\ x(k+2|k) &= A^2x(k|k) + ABu(k|k) + Bu(k+1|k) \\ x(k+3|k) &= A^3x(k|k) + A^2Bu(k|k) + ABu(k+1|k) + \\ &\quad Bu(k+2|k) \\ &\vdots \qquad\qquad \vdots \\ x(k+H|k) &= A^Hx(k|k) + A^{H-1}Bu(k|k) + \\ &\quad A^{H-2}Bu(k+1|k) + \cdots + Bu(k+H-1|k) \end{aligned} \tag{2}$$

By introducing the vector X which contains the set of n state variables to be considered, and the vectors $\hat{Y}$ and U which are the set of future outputs and inputs, respectively

$$\hat{Y} = \begin{bmatrix} y(k+1|k) \\ y(k+2|k) \\ y(k+3|k) \\ \vdots \\ y(k+H|k) \end{bmatrix} \quad U = \begin{bmatrix} u(k|k) \\ u(k+1|k) \\ u(k+2|k) \\ \vdots \\ u(k+H-1|k), \end{bmatrix} \quad X = \begin{bmatrix} x_1(k|k) \\ x_2(k|k) \\ x_3(k|k) \\ \vdots \\ x_n(k|k) \end{bmatrix}$$

Using the generalization of eq. (2), the prediction at sampling instant k is then expressed as

$$\hat{Y} = WU + \Gamma X \tag{3}$$

where the matrices W and Γ are defined as

$$W = \begin{bmatrix} CB & 0 & 0 & \dots & 0 \\ CAB & CB & 0 & \dots & 0 \\ CA^2B & CAB & CB & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ CA^{H-1}B & CA^{H-2}B & CA^{H-3}B & \dots & CB \end{bmatrix}$$

$$\Gamma = \begin{bmatrix} CA \\ CA^2 \\ CA^3 \\ \vdots \\ CA^H \end{bmatrix}$$

Various MPC algorithms propose different cost functions for obtaining the control law. The general aim is that the future outputs ($\hat{Y}$) on the considered prediction horizon must follow a determined reference or setpoint r, and at the same time, the control effort u necessary for doing so should be penalized. The conventional cost function for MPC formulation is written as:

$$J = \sum_{i=1}^{H} \|y(k+i|k) - r(k+i|k)\|_Q^2 \; + \; \|u(k+i-1|k)\|_R^2 \quad (4)$$

In order to obtain the values $u(k+i-1|k)$ in a fixed horizon, where $i = 1, 2, ..., H$, it is necessary to minimize the cost function in (4). Assume that a full measurement of the state $x(k)$ is available at the current time k. Then for event (x, k) (i.e. for state x at time k), a receding horizon-based strategy is implemented by introducing the following optimization problem written on the following form

$$\underset{x_k^{k+H}, u_k^{k+H-1}}{\text{minimize}} \quad J(x(k), u(k)) \quad (5)$$

subject to

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) \\ u(k+i) &\in \mathbb{U}, \quad i \in [0, H-1] \\ x(k+i+1) &\in \mathbb{X}, \quad i \in [0, H-1] \end{aligned} \quad (6)$$

where H is the prediction horizon, $x_k^{k+H} = [x(k+1), x(k+2), .., u(k+H)]^T$, $u_k^{k+H-1} = [u(k), u(k+1), .., u(k+H-1)]^T$, and $J(\cdot)$ is the objective function or the cost function. $\mathbb{U}$ and $\mathbb{X}$ are the sets of admissible values for the control and for the state, respectively. In (5), it tries to minimize the cost function subjected to different constraints found in (6).

In the context this thesis, the constraints are expressed as linear and nonlinear inequalities. The nonlinear constraints are detailed in (III-B) which are intended for obstacle concern. Introducing the vector $Z_k = [\hat{Y}\, U]^T$ that contains the set of future outputs and inputs, the mathematical optimization problem in (5) and (6) can be modified as

$$\begin{aligned} \min_{Z_k} \quad & \|M_y Z_k - \bar{Y}\|_Q^2 \; + \; \|M_u Z_k\|_R^p \\ \text{s.t.} \quad & [I \quad -W] Z_k = \Gamma X \\ & \begin{bmatrix} I \\ -I \end{bmatrix} Z_k \leq \begin{bmatrix} Z_{max} \\ -Z_{min} \end{bmatrix} \\ & h(M_y Z_k) \leq 0 \end{aligned} \quad (7)$$

where $\bar{Y}$ is a set of reference points defined as

$$\bar{Y} = \begin{bmatrix} r(k+1|k) \\ r(k+2|k) \\ r(k+3|k) \\ \vdots \\ r(k+H|k) \end{bmatrix},$$

and M_y and M_u are selection matrices which extract from the vector Z_k the inputs and outputs, respectively. The last two constraints represent the bound for control input and output constraints, and the nonlinear function for obstacle constraints, respectively. Note that the p is either the ℓ_1-norm which is for the sparse controller or ℓ_2-norm which is for the quadratic controller.

After obtaining the control law or the set of optimal values for control, only the first element is applied for the current state. In the following sample, the state is measured again. Afterwards, the whole process is repeated at the next time interval in a receding horizon fashion obtaining a new set of control inputs.

A. Single Integrator

For the linear implementation, the single integrator is used as the plant or vehicle model. In a single integrator model, the motion of the vehicle is given by

$$\dot{x} = u \quad (8)$$

where $\dot{x} = [x_1, x_2]^T$ and $u = [u_1, u_2]^T$ are the control vectors. The discretized equivalent for this model can be defined as

$$\begin{aligned} x_1(k+1) &= x_1(k) + T_s u_1(k) \\ x_2(k+1) &= x_2(k) + T_s u_2(k) \end{aligned} \quad (9)$$

where T_s is the sampling time. Thus, the state-space representation of this model can be written as

$$x(k+1) = \begin{bmatrix} x_1(k+1) \\ x_2(k+2) \end{bmatrix} A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} B = \begin{bmatrix} T_s & 0 \\ 0 & T_s \end{bmatrix} C = A$$

B. Obstacle Constraints

The encoding of obstacle constraints assumes that the shape of every obstacle follows that of the shape of ℓ_p-norms. The geometrical interpretations in 2-D space of these different norms can be shown in Fig. (2).

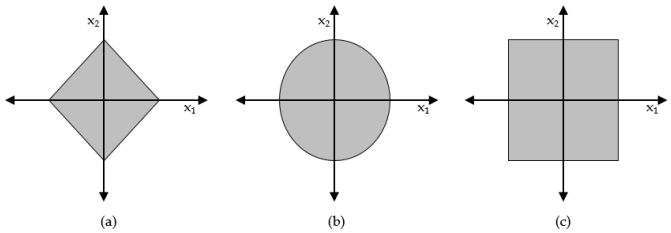

Figure 2. Different Norms Geometric Interpretations in 2D Space. The figures in (a), (b), and (c) correspond to the graphs of ℓ_1, ℓ_2, and ℓ_∞-norms, respectively.

The equation for each of the norms can then be formulated as

$$\|\mathbf{x} - C\|^1 = |x_1 - c_1| + |x_2 - c_2| = d_{safe} \quad (10)$$

$$\|\mathbf{x} - C\|^2 = (x_1 - c_1)^2 + (x_2 - c_2)^2 = d_{safe}^2 \quad (11)$$

$$\|\mathbf{x} - C\|^\infty = max\{|x_1 - c_1|, |x_2 - c_2|\} = d_{safe} \quad (12)$$

Furthermore, the vehicle representation is taken into account in formulating the constraints for a multi-vehicle scenario or it can be interpreted as a moving obstacle. The constraint can be formulated as

$$\sqrt{(x-c_1)^2+(y-c_2)^2} = d_{safe} \tag{13}$$

where d_{safe} is the allowable distance between the vehicle being studied and the vehicle being a moving obstacle with center at C(c_1,c_2).

IV. Nonlinear Sparse Predictive Control

In a nonlinear sparse predictive control formulation, the system is described or approximated by a nonlinear discrete-time model . The vehicle model used for this demonstration is the unicycle model. This model assumes that a vehicle has one steerable drive wheel as shown in Fig. (3). Moreover, this model ignores balancing concerns and it is assumed that the motion of the vehicle cannot slip laterally so that the translational velocity is in the direction of heading, i.e. a pure rolling contact between the wheels and the ground.

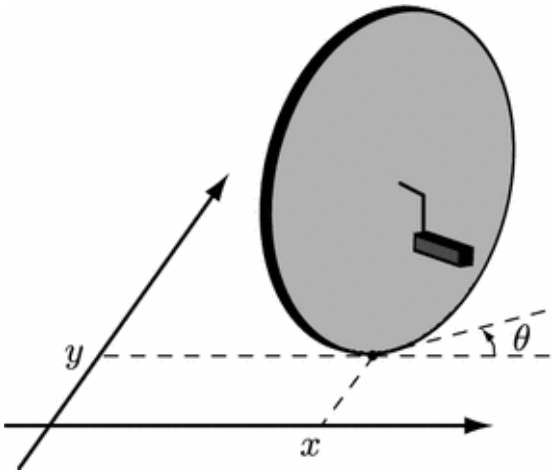

Figure 3. The Unicycle Model (adapted from [15])

The state equations of this model is then given by [16]

$$\begin{aligned} \dot{x} &= v cos\theta \\ \dot{y} &= v sin\theta \\ \dot{\theta} &= w \end{aligned} \tag{14}$$

where state $[x, y, \theta]^T$ describes the position and orientation of the center of the vehicle, and the control inputs denoted by $u = [v, \omega]^T$ are the linear and angular velocities, respectively.

It is important to discretized the model (14) as required in the MPC formulation. Considering a sampling time T_s and a sampling instant k, the discretized equivalent equations for this model can be expressed as

$$\begin{aligned} x(k+1) &= x(k) + T_s v(k) cos\theta(k) \\ y(k+1) &= y(k) + T_s v(k) sin\theta(k) \\ \theta(k+1) &= \theta(k) + T_s w(k) \end{aligned} \tag{15}$$

or, in a compact formulation,

$$\mathbf{x}(k+1) = f_d(\mathbf{x}(k), \mathbf{u}(k)) \tag{16}$$

As proposed in [17], the polar coordinate transformation yields a smoother trajectory and also eliminates the steady-state error effectively. By doing so, the following state transformation to polar coordinates from the Cartesian coordinates is presented, assuming that the reference or goal point is in the origin $(x_g, y_g, \theta_g) = (0, 0, 0)$ and point (x, y) is the center of gravity of the vehicle:

$$\begin{aligned} e &= \sqrt{x_e^2 - y_e^2} \\ \phi &= tan^{-1}(\frac{y_e}{x_e}) \\ \alpha &= \phi - \theta \end{aligned} \tag{17}$$

where x_e = x - x_g and y_e = y - y_g. Letting $\mathbf{X}$ = $[e\ \phi\ \alpha]^T$ and defining a compact formulation as

$$\mathbf{X}(k+1) = f_d(\mathbf{x}(k), \mathbf{u}(k)), \tag{18}$$

By expanding this formulation to the future,

$$\begin{aligned} &\mathbf{X}(k+1|k) = f(\mathbf{x}(k|k), \mathbf{u}(k|k)) \\ &\mathbf{X}(k+2|k) = f(\mathbf{x}(k+1|k), \mathbf{u}(k+1|k)) \\ &\quad\vdots \qquad\qquad\qquad \vdots \\ &\mathbf{X}(k+H|k) = f(\mathbf{x}(k+H-1|k), \mathbf{u}(k+H-1|k)) \end{aligned} \tag{19}$$

the prediction model can be formulated as

$$\mathbf{X}(k+i+1|k) = f_d(\mathbf{x}(k+i|k), \mathbf{u}(k+i|k)), \quad i \in [0, H-1] \tag{20}$$

For a nonlinear sparse controller, the cost function to be minimized can be written as

$$\begin{aligned} J(k) &= \sum_{i=1}^{H} \mathbf{X}^T(k+i|k)\, \mathbf{Q}\, \mathbf{X}(k+i|k) + \mathbf{R}\, |\mathbf{u}(k+i-1|k)| \\ J(k) &= \sum_{i=1}^{H} \|\mathbf{X}(k+i|k)\|_Q^2 \ + \ \|\mathbf{u}(k+i-1|k)\|_R^1 . \end{aligned} \tag{21}$$

On the other hand, the cost function of quadratic controller will take the form

$$\begin{aligned} J(k) &= \sum_{i=1}^{H} \mathbf{X}^T(k+i|k)\, \mathbf{Q}\, \mathbf{X}(k+i|k) + \\ &\qquad \mathbf{u}^T(k+i-1|k) \mathbf{R}\, \mathbf{u}(k+i-1|k) \\ J(k) &= \sum_{i=1}^{H} \|\mathbf{X}(k+i|k)\|_Q^2 \ + \ \|\mathbf{u}(k+i-1|k)\|_R^2 . \end{aligned} \tag{22}$$

For simplicity, both controllers will have the combined form as

$$J(k) = \sum_{i=1}^{H} \|\mathbf{X}(k+i|k)\|_Q^2 \ + \ \|\mathbf{u}(k+i-1|k)\|_R^p \tag{23}$$

where p = 1 is for the sparse controller and p = 2 is for the quadratic controller.

Another cost function to consider in this demonstration is the addition of terminal cost in the objective function in (23), which will be mostly used for the simulations to minimize the convergence error of the controllers. The modified cost function is given by

$$J(k) = \Omega(\mathbf{X}(k+H|k)) + \sum_{i=1}^{H} \|\mathbf{X}(k+i|k)\|_Q^2 + \|\mathbf{u}(k+i-1|k)\|_R^p \tag{24}$$

where $\Omega((\mathbf{X}(k+i|k))(k+H|k)) = \mathbf{X}^T(k+H|k)\ P\ \mathbf{X}(k+H|k)$. The terminal weight factor $P = N*Q$, where N is an arbitrary number.

The mathematical optimization problem can therefore be stated as

$$\begin{aligned} \min_{\mathbf{x},\mathbf{u}} \quad & J(k) \\ \text{s.t.} \quad & \mathbf{x}(k+i+1|k) = f_d(\mathbf{x}(k+i|k), \mathbf{u}(k+i|k)) \\ & \mathbf{u}_{min} \leq \mathbf{u} \leq \mathbf{u}_{max} \\ & h(\mathbf{x}) \leq 0 \end{aligned} \tag{25}$$

where $h(\cdot)$ represents the nonlinear function for the obstacle constraints. The same scheme with the linear MPC still holds to obtain the control law.

V. Results of Nonlinear Sparse Predictive Control Simulations

It is presented in this section the results obtained from the nonlinear controllers implementation. Simulations are obtained from using the $fmincon$ solver in MATLAB to solve for the optimization problem. Throughout this section, the following control variable constraints are chosen as follows:

$$\mathbf{u}_{min} = \begin{bmatrix} -0.47\,m/s \\ -3.77\,rad/s \end{bmatrix}, \quad \mathbf{u}_{max} = \begin{bmatrix} 0.47\,m/s \\ 3.77\,rad/s \end{bmatrix}$$

and the sampling period used is $T_s = 0.1s$.

A. Validations

The first results are for the validations of a working controllers in a free environment, meaning that there are no obstacles present. The weighing matrices used are

$$\mathbf{Q} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0.5 \end{bmatrix}, \quad \mathbf{R} = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix}$$

which means that equal penalties are given to the position states, half of the penalty is imposed in the orientation, and the control effort is penalized less. Throughout the whole validation, the cost function used is in the form of (23), unless stated otherwise.

To begin with, different initial positions are considered: $(x_0, y_0, \theta_0) = a)\,(-2, 2, \frac{\pi}{2})$, $b)\ (2, 2, \frac{\pi}{2})$, $c)\ (2, 0, \frac{\pi}{2})$, $d)\ (2, -2, \frac{\pi}{2})$, $e)\ (-2, -2, \frac{\pi}{2})$, and $f)\ (-2, 0, \frac{\pi}{2})$. Each of the positions is assumed to be occupied by a vehicle. Each vehicle is tasked to reach the set point or the goal point located in the origin $(x_g, y_g, \theta_g) = (0, 0, 0)$, with subscript g denotes the goal point. The trajectory results are shown in Fig. (4) with prediction horizon $H = 5$.

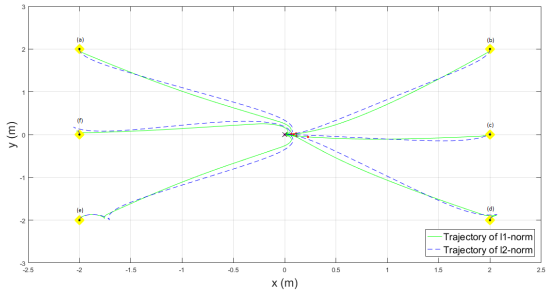

Figure 4. Trajectory plots for the quadratic and sparse controllers using the unicycle model

It can be deduced from Fig. (4) that both controllers satisfy the objective to perform in an obstacle-free environment. It is noticeable, however, that the trajectories of both controllers slightly differ from each other but is understandable. Furthermore, it is visible that some final points (labeled as red dot(·) of the sparse controller do not converge well to the goal point or the origin.

Another interesting thing to investigate is the capability of the controllers to steer regardless of the initial orientations. With this, the initial orientations to be performed are the following: $a)\ \theta_0 = 0$, $b)\ \theta_0 = -\frac{\pi}{2}$, $c)\ \theta_0 = \pi$, and $d)\ \theta_0 = \frac{\pi}{4}$ with the same conditions from the previous case. The results of this demonstration are shown in Fig. (5). It is assumed that both controllers converge to the origin.

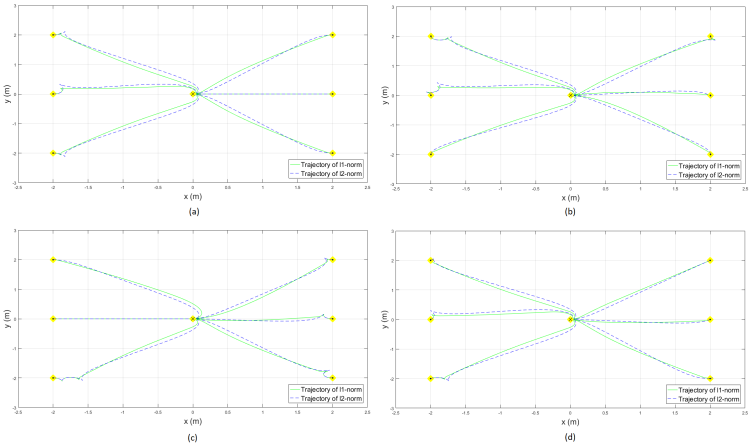

Figure 5. Trajectory plots for the quadratic and sparse controllers using the unicycle model with different initial orientations. Results obtained with the following initial orientations: $a)\ \theta_0 = 0$, $b)\ \theta_0 = -\frac{\pi}{2}$, $c)\ \theta_0 = \pi$, and $d)\ \theta_0 = \frac{\pi}{4}$.

It can clearly demonstrate that the controller developed can performed regardless of the initial positions and orientations.

As noticed earlier in Fig. (4), the convergence of the sparse controller is not so desirable. To minimize the error of

convergence of the sparse controller, it is proposed to increase the prediction horizon. Shown in Fig. (6) and Fig. (7) the results of these changes on both controllers.

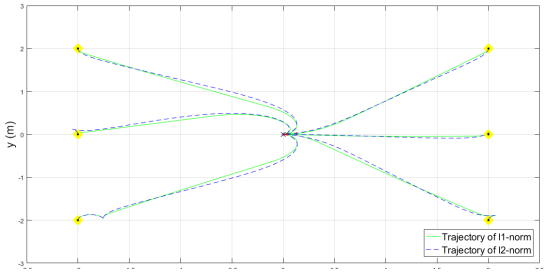

Figure 6. Trajectory plots with H = 10

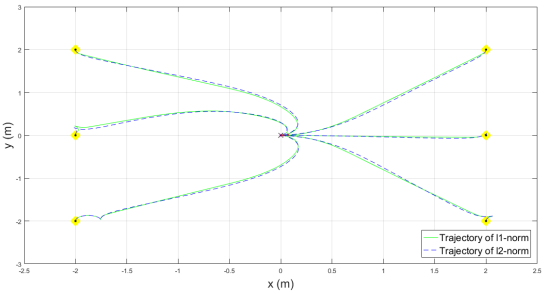

Figure 7. Trajectory plots with H = 20

It can be observed that increasing the prediction horizon will minimize the convergence error of the sparse controller. Furthermore, as the prediction horizon increases, the trajectory of sparse controller tends to behave as the quadratic controller.

In an attempt to reduce or eliminate the convergence error, modification on the cost function is also suggested. This modification is given in (24). For this modification, the weight factor Q remains constant, meaning, it is time-invariant. The terminal state is penalize with a weight factor P = N*Q. For this experiment, different numbers of N are chosen to be simulated. These numbers are: $a)$ $N = 1$, $b)$ $N = 5$, $c)$ $N = 10$, and $d)$ $N = 50$. These values are simulated with H = 10 and the results are all displayed in Fig. (8), considering only the sparse controller.

Notice how the terminal state converges in the origin as the terminal weight factor P is increased by tuning the value of N. It can be seen that at a certain value of N, the trajectory generated will start to change. In Fig. (8d), this change can be observed. Note also that the terminal state of some vehicles for this value of N tends to become unsteady, meaning a fluctuations can be seen as the vehicle tries to settle in the goal point.

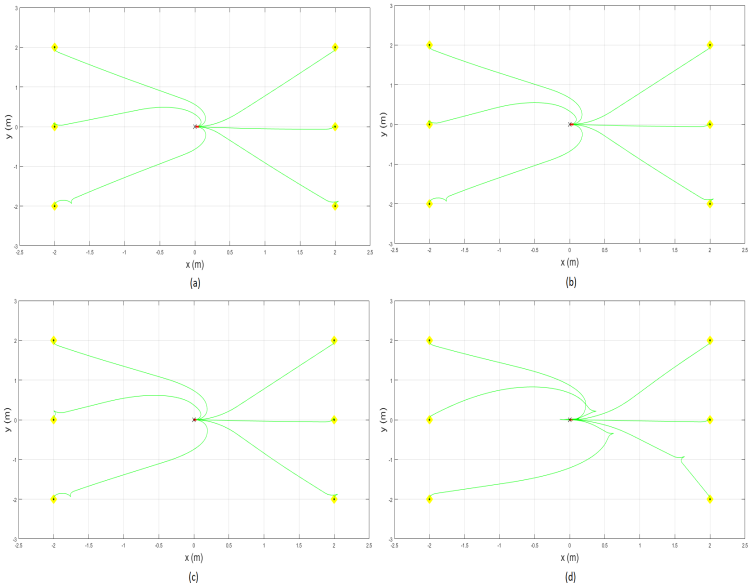

Figure 8. Trajectory plots for a different cost function of sparse controller Results obtained with the following N: $a)$ $N = 1$, $b)$ $N = 5$, $c)$ $N = 10$, and $d)$ $N = 50$.

Though it is verified in here that adding a terminal cost will indeed improve the convergence of the terminal state, recall that a careful choice of weight factors, not only the terminal weight factor, is still a concern. Nevertheless, the addition of terminal cost is recommended in the context of this thesis to solve the issue in convergence error for the nonlinear sparse controller, and hereby be carried out in the remainder of this section.

B. Obstacle Avoidance Problem

The final results are intended for a situation where obstacles are considered in the environment. The vehicle is simulated to start at point (x_0, y_0, θ_0) = $(-2, 2, \frac{\pi}{2})$ and is tasked to steer at point (x_g, y_g, θ_g) = $(0, 0, 0)$. For this demonstration, a circular-shape obstacle is considered and is centered at point (x_b, y_b) = $(-1, 1)$. Fig. (9) shows the trajectories for both quadratic and sparse controllers utilizing the cost function in (24) with parameters: prediction horizon H = 20, and weight matrices R = diag(1,1), Q = diag(1,1,0.5) and $P = 20 * Q$.

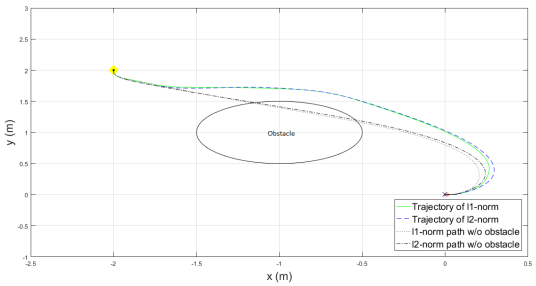

Figure 9. Trajectory for sparse and quadratic controllers with obstacle.

As shown, the sparse controller is able to handle the obstacle constraint, the same with the quadratic controller. The trajectories with the obstacle can be described as the path taken by the vehicle when it traverses the environment. Upon detecting the obstacle in its way to the origin, it keeps a distance away from the it and changes its direction of heading.

The cost over time is shown in Fig. (10), while the control effort and individual state plots are shown in Fig. (11).

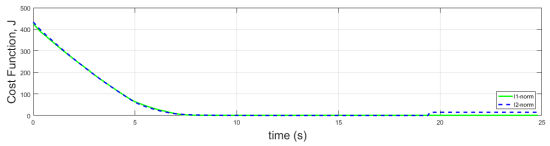

Figure 10. Cost over time for a higher penalty on the control cost.

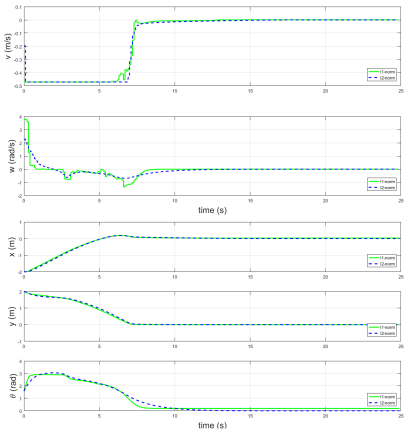

Figure 11. Control effort and state plots for a higher penalty on the control cost.

The total cost calculated from (10) for the sparse controller is 12500 while 13300 is calculated for the quadratic controller. With these figures, it can be said that the sparse controller minimizes more the cost incurred during the whole process with a satisfactory performance comparable to the quadratic controller.

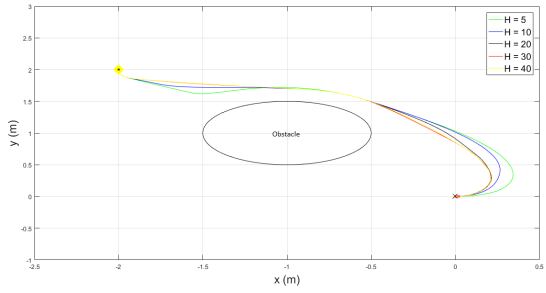

Figure 12. Trajectories of different horizons in obstacle avoidance.

To demonstrate the importance of the prediction horizon in obstacle avoidance scenario, different values of horizon lengths are considered. Fig. (12) depicts the result of performing different horizons with weight matrices Q = diag(1,1,0.5), R = diag(0.1,0.1), and $P = 20 * Q$. The figure implies that larger prediction horizons are required to ensure a smooth maneuver by giving the vehicle more information on a look ahead basis, thus planning its path earlier upon detecting the obstacle.

For a multiple obstacles demonstration, assumed that the environment is occupied by 2 obstacles represented as circles. The resulting trajectory is shown in Fig. (13).

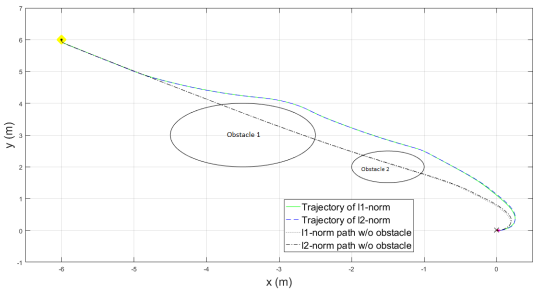

Figure 13. Trajectories of both controllers in obstacle-free and multiple obstacles problem.

The trajectories of both controllers in a free environment are also provided. As it can be seen, both controllers can also be applied in a multiple obstacles scenario where the obstacles are represented as a disc or circle. The control efforts of both controllers in multiple obstacles are revealed in Fig. (14).

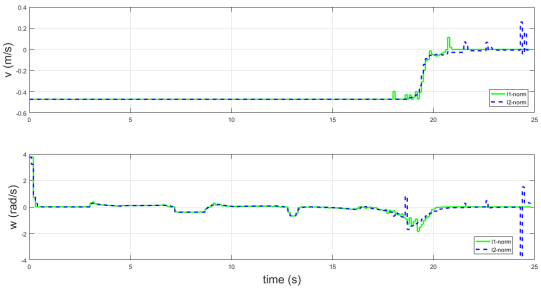

Figure 14. Control Effort of both controllers for multiple obstacles problem.

It is seen that the control effort for sparse controller has more stabilized region than the quadratic one. This stabilized region can be illustrated by the nature of the ℓ_1-norm penalty or sparsity, where it penalizes the number of nonzero elements and force many elements to be equal to zero. As a consequence, it can also be noticed that fluctuations of control values are minimal for sparse controller compare to the quadratic controller.

C. Evaluation of the Effect of Parameters on the Cost

The results presented in this section are done in an environment with one obstacle just like the environment as shown in Fig. (9). The purpose of this section is to present the results

of fine tuning significant parameters such as the prediction horizon, and weight matrices.

To check the effect of prediction horizon to the cost function of nonlinear sparse predictive control algorithm, the following fixed parameters are used as shown in Table (I).

Parameter	Value
Q	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0.5 \end{bmatrix}$
R	$\begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix}$
P	20*Q

Table I
Fixed Parameters for Cost vs. H evaluation.

Different prediction horizons are considered to determine its corresponding total cost. These horizons are the following: $H = 10$, $H = 20$, $H = 30$, and $H = 40$. Fig. (15) depicts the cost over time of the horizon choices.

For the purpose of this evaluation for the effect of horizons on the cost of both controllers, fixed parameters assigned remain as is. In the figure, the trajectories of the cost can be seen as exponentially decreasing. The corresponding total cost incurred by both controllers in each defined horizon is also shown. It can be inferred that increasing the prediction horizon will make the process more costly. However, for this demonstration at a certain point, the increase of the total cost is only minimal or is not clearly seen. This can be justified by the minimization problem defined. Since the aim here is to approach the origin, the values in higher horizons are smaller ones thus these values do not really contribute to the totality of cost.

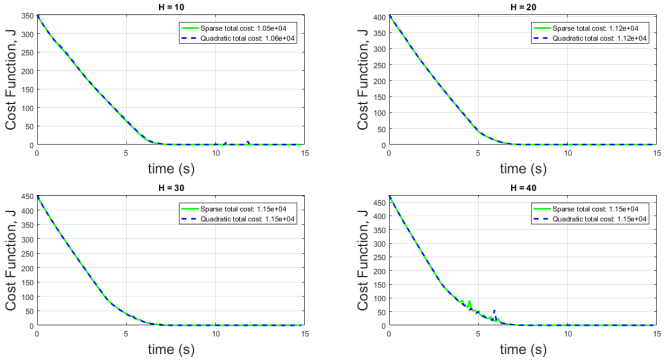

Figure 15. Cost over time of different horizons used.

To determine the effect of weight matrices Q, R, and P to the cost function of nonlinear sparse predictive control algorithm, the prediction horizon is fixed at H = 20. Only one case per weight matrix will be considered. It is assumed that a higher value on every weights is enough to learn the significant effect of these weights.

For the effect of weight factor Q, which penalizes the deviation of the state from the goal point, the following weight matrices are considered: $Q = \begin{bmatrix} 10 & 0 & 0 \\ 0 & 10 & 0 \\ 0 & 0 & 0.5 \end{bmatrix}$, $R = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix}$ and P = 20*Q. Fig. (16) shows the comparison of the cost over time between the result as seen in Fig. (15) with H = 20 and with the new defined matrix Q.

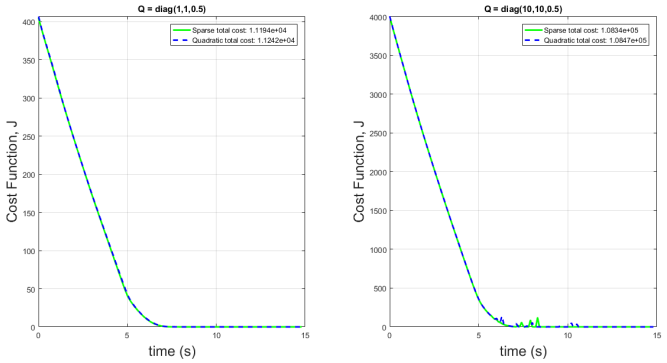

Figure 16. Cost over time of the change of matrix Q.

It can be seen that a huge difference has been calculated as a consequence of penalizing the position about 10 times higher. The difference can also be approximated as 10 times more of the former result.

Evaluating the effect of weight factor R, which penalizes the control effort, the following weight matrices are considered: $Q = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0.5 \end{bmatrix}$, $R = \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix}$ and P = 20*Q. Again, the only change from Fig. (15) is the weight factor R with 5 times the penalty now on the inputs. Comparison result is shown in Fig. (17).

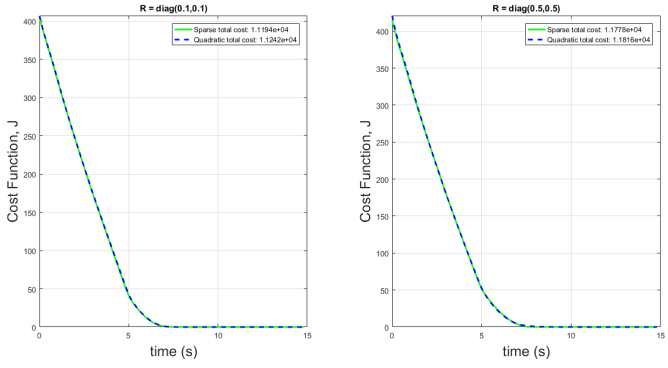

Figure 17. Cost over time of the change of matrix R.

Comparing the values shown in the figure, a slight change occurs. This can be analyzed by recalling the constraints on the control inputs, which are only small values. Furthermore, even though the change is 5 times of the matrix R in Fig. (15), it is still half the penalty(0.5) of a whole. For demonstration

purposes, the only change is with matrix R and increasing this penalty will result to inability of the vehicle to steer.

Since the cost function used also has an additional terminal cost with weight factor P, the following weight matrices are used for evaluation: $Q = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0.5 \end{bmatrix}$, $R = \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix}$ and $P = 40*Q$. Again, the only change from Fig. (15) is the weight factor P with two times the penalty now on the terminal cost. Result is shown in Fig. (18).

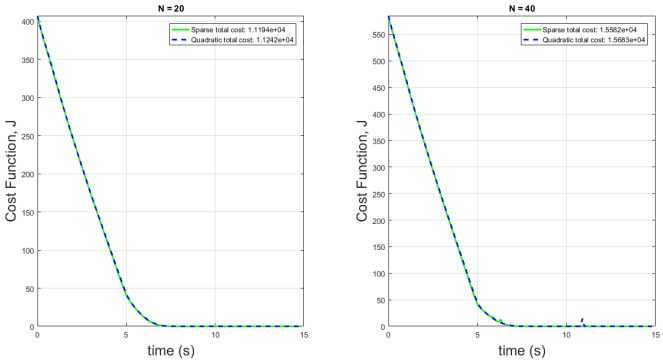

Figure 18. Cost over time of the change of matrix P.

Comparing the values indicated in the figure, a big difference is also observed in relation to the change in matrix Q previously. This increase can be explained with penalizing the quadratic nature of the terminal cost.

VI. Results of Linear Sparse Predictive Control Simulations

This section shows the results obtained from using a linear system in a defined environment with different types of obstacles. The optimization problem is still solved using the $fmincon$ solver in MATLAB. After the selection of parameters process, the following parameters are chosen for the demonstration: H = 10, T_s = 0.1s, Q = diag(20,1), R = diag(10,1). The control inputs are bounded by $-5 \leq u_i \leq 5$ in every iteration. Furthermore, to make the movement of the vehicle more realistic, the position is constrained by

$$\begin{bmatrix} x_{k-1} - 2 \\ y_{k-1} - 2 \end{bmatrix} \leq \begin{bmatrix} x_k \\ y_k \end{bmatrix} \leq \begin{bmatrix} x_{k-1} + 2 \\ y_{k-1} + 2 \end{bmatrix}$$

The output variables are represented as $[x, y]^T$, which is equivalent to the states $[x_1, x_2]^T$. Resulting trajectory is shown in Fig. (19). Note that the trajectory of the sparse controller is adjusted so it will look like settling in the goal perfectly. In real simulation, the trajectory produced by sparse controller does not converge in the goal point (with minimal gap only).

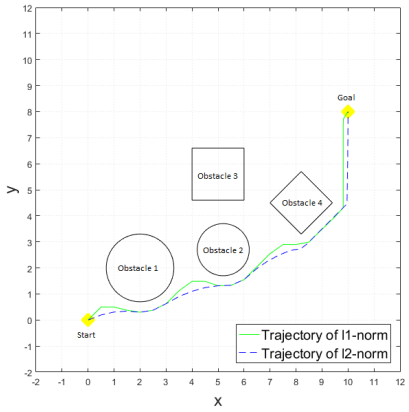

Figure 19. Position plots for the quadratic and sparse controllers

Results for the control inputs of both controllers are presented in Fig. (20) while the results for the outputs are shown in Fig. (21).

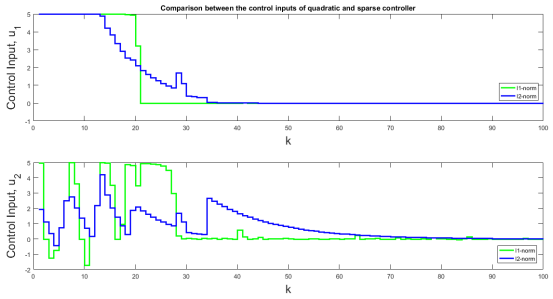

Figure 20. Control inputs plots for the quadratic and sparse controllers

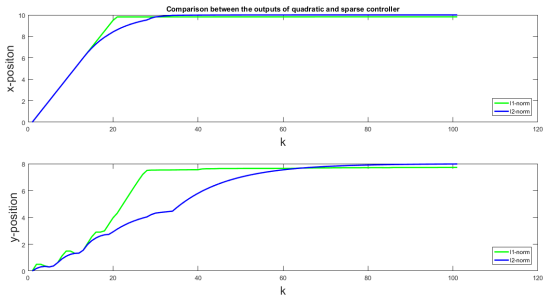

Figure 21. Position plots for the quadratic and sparse controllers

The difference between the two controllers can be shown in different trajectories produced. In sparsity, it is expected that the control set will contain few non-zero elements. Even if it is applied in a receding horizon method, the characteristic of sparsity will still be visible as a whole. As shown in Fig. (20), the control inputs' trajectory of sparse controller tends to immediately cut the control as opposed to a decaying trend of a quadratic controller at a later time, k. This aspect of sparse controller is more desirable than the quadratic controller since it minimizes more the control effort, thus yielding a better fuel performance if it relates to vehicles. However, this difference

is not so evident if one looks into the position plot in Fig. (21). But by looking into the y-position, one can justify that at some point, the sparse-controlled vehicle will accelerate more than the quadratic-controlled one.

Solving for the total cost of the sparse controller in the presented example, the following result is provided:

$$J_{\ell_1} = \sum_{k=0}^{N=100} (\hat{y}_{k+i} - r_i)^T Q(\hat{y}_{k+i} - r_i) + R|u_{k+i-1}| \\ = 9.56\mathbf{x}10^4 \tag{26}$$

Furthermore, the total cost for the quadratic controller is calculated as

$$J_{\ell_2} = \sum_{k=0}^{N=100} (\hat{y}_{k+i} - r_k)^T Q(\hat{y}_{k+i} - r_k) + (u_k)^T R(u_k) \\ = 1.11\mathbf{x}10^5 \tag{27}$$

Comparing the result, notice that the quadratic controller is more costly than the sparse controller by a substantial percentage or margin of

$$\%difference = \frac{J_{\ell_2} - J_{\ell_1}}{J_{\ell_1}} \times 100 = 16.1\%$$

VII. Conclusion

The aim of this thesis is to develop a control strategy utilizing the Model Predictive Control (MPC) scheme and sparsity for vehicles modeled as linear, as well as nonlinear systems. MPC is a control strategy that generates the control action by solving an optimization problem in a finite receding horizon principle, with the measured state as the initial state, that can handle the state and input constraints directly. The main components of the MPC algorithm include the performance measure or the cost function. In this work, the conventional cost function of MPC is modified by applying the idea of sparsity, which penalizes the the number of nonzero elements in the control sequence. This modification resulted to what is known, in the context of this thesis, as the sparse predictive control strategy.

The results suggest that the sparse predictive control developed is better, if not as good as the quadratic controller. Though a significant issue is found in sparse controller for its incapacity to reach the goal point as desired, but this drawback is addressed by adding a terminal cost in the objective function. Fine tuning of parameters will also help solve the problem without affecting the solution of the sparse controller, but the parameters chosen might not be suitable for the quadratic controller for comparison purposes. The result for the sparse controller is shown to stabilize more the movement of the vehicle after reaching the goal point. The control effort in the entire process or journey is able to demonstrate the nature of sparsity, thus believing to minimize the fuel consumption if it relates to a real world scenario.

References

[1] R. E. Kalman, "Contributions to the Theory of Optimal Control," *Bulletin de la Societe Mathematique de Mexicana*, no. 5, pp. 102-119, 1960.

[2] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," *Transactions of ASME, Journal of Basic Engineering*, no. 87, pp. 35-45, 1960.

[3] S. J. Qin and T. A. Badgwell, "A Survey of Industrial Model Predictive Control Technology," 2002, available at https://www.sciencedirect.com/science/article/pii/S0967066102001867

[4] J. Richalet, A. Rault, J. Testud, and J. Papon, "Model Predictive Heuristic Control: Applications to Industrial Processes," 1978, available at https://www.sciencedirect.com/science/article/pii/0005109878900018

[5] C. Cutler and B. Ramaker, "Dynamic Matrix Control - A Computer Control Algorithm," *Automatic Control Conference*, 1980.

[6] J. M. Maciejowski, *Predictive Control with Constraints*. Pearson Education Limited, 2002.

[7] D. W. Clarke, C. Mohtadi, and P. Tuffs, "Generalized Predictive Control-Part I. The Basic Algorithm," *Automatica*, vol. 23, no. 2, pp. 137-148, 1987.

[8] D. W. Clarke, C. Mohtadi, and P. Tuffs, "Generalized Predictive Control-Part II. Extension and Interpretations," *Automatica*, vol. 23, no. 2, pp. 149–160, 1987.

[9] W. Kwon and S. Han, *Receding Horizon Control Model Predictive Control for State Models*. Springer-Verlag London Limited, 2005.

[10] J. Mairal, F. Bach, and J. Ponce, "Sparse Modeling for Image and Vision Processing," *Foundations and Trends in Computer Graphics and Vision*, vol. 8, no. 2-3, pp. 85–283, 2014.

[11] M. Gallieri and J. M. Maciejowski, "Stabilising Terminal Cost and Terminal Controller for ℓ_{asso}-MPC: Enhanced Optimality and Region of Attraction," *European Control Conference (ECC)*, pp. 524–529, July 17-19, 2013.

[12] S. K. Pakazad, H. Ohlsson, and L. Ljung, "Sparse Control Using Sum-of-Norms Regularized Model Predictive Control," *52nd IEEE Conference on Decision and Control*, pp. 5758–5763, December 10-13, 2013.

[13] L. Dai, Y. Xia, M. Fu and M. S. Mahmoud, "Discrete-Time Model Predictive Control, Advances in Discrete Time Systems Magdi Mahmoud, IntechOpen, DOI: 10.5772/51122," December 5, 2012, available at https://www.intechopen.com/books/advances-in-discrete-time-systems/discrete-time-model-predictive-control

[14] D. E. Seborg, T. F. Edgar, and D. A. Mellichamp, *Process Dynamics and Control*, 2nd ed. John Wiley & Sons, Inc., 2004.

[15] B. A. Francis and M. Maggiore, *Flocking and Rendezvous in Distributed Robotics*. Springer International Publishing, 2016.

[16] S. M. LaValle, *Planning Algorithm*. Cambridge University Press, 2006.

[17] F. Kuhne, W. F. Lages, and J. M. G. da Silva Jr., "Point Stabilization of Mobile Robots with Nonlinear Model Predictive Control," *IEEE International Conference on Mechatronics and Automation*, pp. 1163–1168, July 2005.

[18] J. B. Rawlings and D. Angeli and C. N. Bates, "Fundamentals of Economic Model Predictive Control," *51st IEEE Conference on Decision and Control*, pp. 3851-3861, December 10-13, 2012.