Control and Obstacle Avoidance of Wheeled Mobile Robot

Muhammad Farhan Manzoor* and Qinghe Wu
School of Automation
Beijing Institute of Technology
Beijing, China
farhan_manzoor@hotmail.com

***Abstract*—The idea belong to presented work is to use a MPC (Model Predictive Control) controller combined with the feedback linearization in order to manage the control problem of a single robot with unicycle Kinematics and obstacle avoidance function. To guarantee the obstacle avoidance of wheeled mobile robot we use the widely popular concept of artificial potential field that perfectly integrates with MPC. This approach can be naturally used for defining a proper cost function in the MPC framework. The proposed algorithm is based on the concept of robust MPC control that guarantees the single robot to avoid collision in the presence of only fixed obstacles. The proposed control algorithm has been validated using a simulation environment. Some of the most relevant results are presented in the simulation section. In the end some possible future research directions are underlined.**

Keywords—model predictive control; feedback linearization; kinemtics; obstacle avoidance; potential field

I. INTRODUCTION

Over a past decade MPC is used to address a wide range of real world problems. This success is motivated due to deal easily with constraints on both state and control variables during the evolution of the system under control. MPC has also shown to be useful particularly for developing distributed multi-agent coordination algorithms, predictions of the state trajectory are available at each time step. The advantage of distributed control with respect to centralized control implementations lies on the fact that the computational load can be distributed among the involved agents.

Many obstacle avoidance algorithms have been developed in the literature that allow for fast, continuous, and smooth motion of wheeled mobile robots among unpredicted obstacles. They can be classified into four categories, graph-search based methods [1] artificial potential field based methods [2] meta-heuristic based methods [3], [4] and mathematical optimization based methods [5]. Among these categories mathematical optimization based methods are particularly attractive, because they offer a rigorous and systematic way to take vehicle dynamics and safety constraints into account and can generate optimal control inputs. A mathematical optimization approach can be used either in an open-loop, if the environment is fully known a priori, or in a closed-loop with a feedback controller for a more robust solution regarding to the MPC. Prior research has demonstrated successful applications of MPC to obstacle avoidance in UGVs [6]. The applications of Nonlinear MPC to ground vehicles is illustrated in [7]. A stability control system using MPC was considered by Anderson, et al. [8].

The objective of this research is to get desired behaviors with low computational cost easily implemented on local agents. In recent years, research in this field is focused in developing optimization problems that ensure robustness of the system in respect to external disturbances concerned the model and the real system, leading to control technique said Robust MPC [9]. Another branch of research in Robust MPC has adopted distributed predictive control (DPC) techniques [10] that includes DPC algorithm, [11] which aims to divide the complete optimization problem into smaller problems, which are locally resolvable more easily and fast.

II. MOBILE ROBOT KINEMATICS

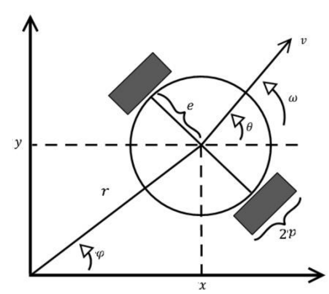

Figure 1. Unicycle Model

The unicycle model is presented in Figure 1 and kinematic equations that govern the motion of the robot is simply as given below

$$\begin{cases} \dot{x} = v\cos\theta \\ \dot{y} = v\sin\theta \\ \theta = \omega \end{cases} \tag{1}$$

Slipping prevention in the cross direction with existing constraint, be able to obtain

$$\dot{y}\cos\theta - \dot{x}\sin\theta = 0 \tag{2}$$

Consider the accelerations along two axes in matrix form we have

978-1-4673-7016-5/15 $31.00
DOI 10.1109/CICSyN.2015.48

$$\begin{pmatrix} a_x \\ a_y \end{pmatrix} = \begin{bmatrix} \cos\theta & -v\sin\theta \\ \sin\theta & v\cos\theta \end{bmatrix} \begin{pmatrix} \dot{v} \\ \omega \end{pmatrix}$$

If $v \neq 0$ the transformation is invertible and we obtain the first order dynamic linearization controller as given

$$\begin{cases} \dot{v} = a_x \cos\theta + a_y \sin\theta \\ \omega = \dfrac{-a_x \sin\theta + a_y \cos\theta}{v} \end{cases} \quad (3)$$

By applying this controller to unicycle the linear and desired feedback system is

$$\begin{cases} \ddot{x} = a_x \\ \ddot{y} = a_y \end{cases} \quad (4)$$

The general control scheme is presented in Figure 2.

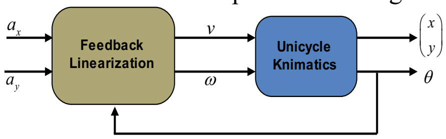

Figure 2. Control Scheme

III. PROBLEM FORMULATION

A. Obstacle Avoidance Formulation

MPC is an optimal control-based state-feedback control technique. Figure 3 illustrates a high level configuration that show how an MPC is utilized in this work for obstacle avoidance purpose. The goal of this paper is to build a control technique which drive the robot to the desired position without a phase of planning the entire route.

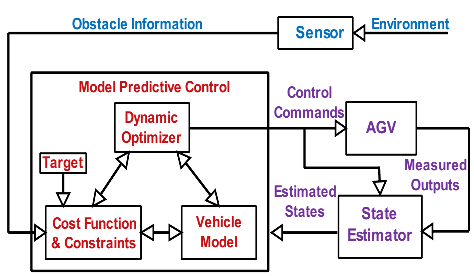

Figure 3. Schematic of the MPC-based obstacle avoidance algorithm

The idea is to use an appropriate local control loop to manage the movement of the robot at a precise reference and use an external control loop based on predictive techniques to generate the best orientation towards the completion of the task, remaining at a distance from obstacles. In this regard the predictive approach for the optimization problem is used, limited horizon considered, obstacle avoidance problems typical of path planning phase of any autonomous navigation problem in the generic sense is considered. It should be noted that the execution of the same task is affected by uncertainty and therefore system must be used in feedback control mechanisms.

B. MPC Problem Formulation

The receding horizon control technique is described in detail in [12] and is currently being considered in this paper. The philosophy of receding horizon is, at time t to solve an optimal control problem over a finite future horizon of steps N and at time $t+1$ acquire future measurements, repeat the optimization and so on. The classical form for the control of discrete-time linear systems used as a reference with state-space representation

$$\begin{cases} x(t+1) = Ax(t) + Bu(t) \\ y(t) = Cx(t) \end{cases} \quad (5)$$

When $x(t) \in R^n$, $u(t) \in R^m$ & $y(t) \in R^p$ respectively, are the state, the control sequence, and the output of the system on the time instant t. $A, B \& C$ are the system, input, and output matrix. This system is assumed to be subject to constraints on permissible values of states and inputs. In particular $x(t) \in \mathrm{X}$, $u(t) \in U$ where U and X is a convex sets, system is assumed to be subject to constraints i.e. without loss of generality and holding the origin in its inner.

Considering the open loop control problem, the purpose is to drive the system state toward the origin within an optimal way satisfying the constraints along the entire interval. The value function V^n that characterizes this optimization problem is generically defined as

$$V^n(x(t), U_t) = \sum_{k=t}^{t+N-1} l(x(k), u(k)) + V^f(x(t+N)) \quad (6)$$

Where U_t indicates the sequence of inputs along the predictive horizon $\{u(t)...u(t+N\text{-}1)\}$. In the V^n expression the term N is the predictive horizon and the elements of the state vector that satisfy the dynamic equation $x(t+1) = Ax(t) + Bu(t)$. A positive definite cost function of the state and input variable is $l(x(k), u(k))$. The cost function in quadratic form is specified as

$$l(x(k), u(k)) = \|x(k)\|_Q^2 + \|u(k)\|_R^2 = x(k)^T Qx(k) + u(k)^T Ru(k)$$

Q And R are positive definite, which applied respectively on state and control. The terminal cost function V^f is also a positive definite function and is given as

$$V^f(x(t+N)) = \|x(t+N)\|_P^2 = x(t+N)^T Px(t+N) \quad (7)$$

With a matrix of weight P belong to the final state. This concept is linked to the function of secondary law or stabilizing law

$$u(k) = kx(k)$$

In addition to the external constraints the affiliation of state and control variable is X and U, need to add the terminal constraint in the form

$$x(t+N) \in X^f \subseteq X \tag{8}$$

The final form $x(t+N)$ attained by the optimizer, falls within a set X^f also centered at the origin. The sequence of values for the control variable is then obtained by minimization of cost function in the presence of constraints

$$U_t^* = \{u(t|t), \ldots u(t+N-1)|t)\} = \\ \arg\min_{U_t} \left(V^N(x(t), U_t), \begin{pmatrix} x(k) \in X \\ u(k) \in U \\ x(t+N) \in X^f \end{pmatrix} \right) \tag{9}$$

This solution must be calculated at each instant of time after which the sequence of first element, $u(t|t)$ is applied to the system. The optimization based on the updated information that can be interpreted finally as a control law, implicit and time invariant and summarized as

$$k_N(x(t)) = u(t|t)$$

By applying this closed loop dynamics becomes

$$x(t+1) = Ax(t) + Bk_N(x(t))$$

In order to discuss the properties of MPC, it is necessary to introduce the notion of positive invariant sets that are described in [13]. Let we have a Positive Invariant(PI) set Ω for an autonomous system $x(t+1)=f(x(t))$ each initial state contained in Ω, the solution $x(t)$ remains confined in Ω for all successive instants of time.

C. Conditions for stability.

The stability of MPC can be proven using Lyapunov criterion and the cost function V^{N^*} produced by optimization. First of all the states within the terminal set must satisfy the constraints, i.e. X^f must be a closed set containing the origin such that $\mathrm{X}^f \subseteq \mathrm{X}$. If the system is controlled through the secondary law the secondary input calculated by the local law $k^f(x)$ must reside in U for all $\mathrm{x} \in \mathrm{X}^f$ [14]. Secondly X^f must be positively invariant with respect to the secondary law $k^f(x)$ presented

$$x(k+1) = Ax + Bk^f(x(k)) \in \mathrm{X}^f \quad \forall x(k) \in \mathrm{X}^f \tag{10}$$

The final cost function V^f must contain a Lyapunov function for the system that has to be controlled through the secondary law chosen as

$$V^f(Ax(k) + Bk^f x(k))) \le V^f(x(k)) - \\ l(x(k), k^f(x(k))) \quad \forall x(k) \in \mathrm{X}^f \tag{11}$$

D. Design choices.

The stability of the system and the properties of MPC exist significantly in the choice of the final cost function V^f. Two strategies adopted with regard to the behavior towards the final state. The first one is assume that the control law is of stabilizing controller $u(k)=0$, the initial system is stable and $(A+BK)$ matrix hold eigenvalues in the unit circle with $K=0$. The second solution refers to the use of secondary stabilizing control law $u(k)=Kx(k)$ with a gain chosen in such a way that $(A+BK)$ is asymptotically stable. The infinite horizon is chosen as $V^f = V_\infty^*$ i.e. obtain its optimal value when $N \to \infty$. Due to the existing constraints problem the value V_∞^* is unidentified.

$$\begin{aligned} \tilde{V}_\infty(x) &= \min_U \sum_{k=0}^{\infty} l(x(k), u(k)) \\ &= \min_U \sum_{k=0}^{\infty} \|x(k)\|_Q^2 + \|u(k)\|_R^2 \end{aligned} \tag{12}$$

V^f comprises the equivalent cost of unbounded infinite horizon $\tilde{V}_\infty$. The solution of this problem is nothing more than the classic discrete-time LQR and the optimum value function can be calculated as

$$\tilde{V}_\infty^*(x) = \|x\|_P^2 = x^T P x \tag{13}$$

In Eq.13 the matrix P denotes the unique positive definite solution of the algebraic Riccati equation in discrete time

$$P = Q + A^T(P - PB(R + B^T PB)^{-1} B^T P)A$$

The optimal controller is linear and consists of a gain K usually referred to as Kalman gain, which can be calculated simply as

$$K = -(R + B^T PB)^{-1} B^T PA$$

A practical choice is to first determine K^f to assign the eigenvalues and then calculate P from discrete Lyapunov formulation

$$(A+BK)^T P(A+BK) - P = (Q + K^T RK)$$

In case where the matrix A is stable the simplest choice is to put $K=0$ and therefore the above equation simplifies and the calculation of P is obtained as

$$A^T PA - P = -Q \tag{14}$$

So as to acquire a region of attraction as wide as desirable for the MPC controller, X^f is selected maximum positive invariant (MPI) of the closed loop system, secondary law is

$$x(t+1) = (A+BK)\mathrm{x(t)} \in \mathrm{X} \tag{15}$$

If the cost functions along the horizon and terminal functions are chosen according to the given guidelines, then the resulting cost function appears $V^N(x(t), u(t:t+N-1|t))$.

The weights of matrices $Q \& R$ on state and control variable and terminal P are easily obtained by solving offline algebraic Riccati equation. In the case the sets $\mathrm{X}, U \& \mathrm{X}^f$ are chosen which can be represented as the intersection of a finite set of subspaces. The resulting optimization problem is still quadratic type and relatively easy to solve. This corresponds for each polyhedral set that describes permissibility constraints $\mathrm{X}, U \& \mathrm{X}^f$. The matrix representation defined as

$$\mathrm{X} = \{x : H_x x \le K_x\} \quad U = \{u : H_u u \le K_u\} \quad \mathrm{X}^f = \{\mathrm{x}_f : H_f x_f \le K_f\}$$

It follows a problem of quadratic programming (QP) which can be solved with a low computational cost by using the modern optimization algorithms. This aspect allows to apply this technique to systems with reduced sampling times.

IV. CONTROL ALGORITHM FOR STABILIZATION

Consider the unicycle discrete-time system obtained from the implementation of feedback linearization as described in section II

$$\begin{cases} \dot{x} = v\cos\theta \\ \dot{y} = v\sin\theta \to \\ \theta = \omega \end{cases} \begin{cases} z_1^+ = z_1 + \tau z_3 \\ z_3^+ = z_3 + \tau a_x \\ z_2^+ = z_2 + \tau z_4 \\ z_4^+ = z_4 + \tau a_y \end{cases} \quad z = \begin{bmatrix} x \\ v_x \\ y \\ v_y \end{bmatrix} \qquad (16)$$

Chosen sampling time obtained by the linear system in matrix form is given as

$$\xi(t+1) = \begin{bmatrix} 1 & \tau & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \tau \\ 0 & 0 & 0 & 1 \end{bmatrix} \xi(t) + \begin{bmatrix} 0 & 0 \\ \tau & 0 \\ 0 & 0 \\ 0 & \tau \end{bmatrix} u(t)$$

Which is comprised of two decoupled subsystems respectively representing the dynamics along the x-axis and the Cartesian co-ordinate of the reference system.
To verify the validity of the approach by linear law that indeed serves as secondary law in MPC problem. In particular we have seen

$$u(t) = \begin{bmatrix} -a_1 & -a_3 & 0 & 0 \\ 0 & 0 & -a_2 & -a_4 \end{bmatrix} \xi(t) = k_{aux}\xi(t) \quad (17)$$

In Eq. 17 the parameter k_{aux} are appropriate gain stabilizers. The MPC control problem is formulated as optimization cost of figure V^n on a finite horizon of length N illustrated as

$$V^N = \sum_{k=t}^{t+N-1} \|\xi(k)\|_Q^2 + \|u(k)\|_R^2 + \|\xi(\mathrm{t}+N)\|_P^2$$

The weights Q, R (and consequently P) chosen as diagonal block with decoupling dynamics obtained by the system. The choice of weight of matrix P and the final status of the optimization on instant $t+N$ simply use the secondary law

$$u(k) = k_{aux}\xi(t) \to \quad \xi(k+1) = (A+B)k_{aux}\xi(\mathrm{k}) \qquad (18)$$

And assuming that the dynamic matrix obtained is stable use the following equation to calculate P

$$(A+BK_{aux})^T P (A+BK_{aux}) - P + (Q + K_{aux}^T R K_{aux}) = 0$$

With the current status of $\xi(t)$, we can write the prediction horizon along the $t+k$ instants, while using $u(t:t+N-1)$ getting optimization variable function

$$\begin{aligned} &\xi(t) = I\xi(t) \\ &\xi(t+1) = A\xi(t) + Bu(t) \\ &\xi(t+2) = A\xi(t+1) - Bu(t+1) = A^2\xi(t) + ABu(t) + Bu(t+1) \\ &\xi(t+k) = A^k\xi(t) + A^{k-1}Bu(t) + \dots AB\,u(t+k-2) + Bu(t+k-1) \end{aligned}$$

The entire prediction horizon along the $t...t+N$ be summed in the following way

$$\underbrace{\begin{bmatrix} \xi(t) \\ \xi(t+1) \\ \vdots \\ \xi(t+N) \end{bmatrix}}_{\Xi_t} = \underbrace{\begin{bmatrix} I \\ A \\ \vdots \\ A^N \end{bmatrix}}_{A} \xi(t) + \underbrace{\begin{bmatrix} 0 & 0 & \cdots & 0 \\ B & 0 & \cdots & 0 \\ \vdots & \vdots & \cdots & \vdots \\ A^{N-1}B & A^{N-2}B & \cdots & B \end{bmatrix}}_{B} \underbrace{\begin{bmatrix} u(t) \\ u(t+1) \\ \vdots \\ u(t+N-1) \end{bmatrix}}_{U_t}$$

Rewriting figure of merit in a compact way by removing the summation. The Ξ_t term consist of $N+1$ elements while U_t is the term which we want to optimize is a component of N vector.

$$V^N = \|\Xi_t\|^2_{\begin{bmatrix} Q & \cdot^{\cdot^{\cdot}} \\ \ddots & P \end{bmatrix}} + \|U_t\|^2_{\begin{bmatrix} R & \cdot^{\cdot^{\cdot}} \\ \ddots & R \end{bmatrix}} = \|A\xi(t) + BU_t\|_{\mathbb{Q}}^2 + \|U_t\|_{\mathbb{R}}^2$$

The state and control input weights are now made up of the matrices given below

$$\mathbb{Q} = \begin{bmatrix} Q & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & P \end{bmatrix} \quad \mathbb{R} = \begin{bmatrix} R & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & R \end{bmatrix}$$

To calculate the minimum optimization function with respect to the vector $U_t = \{u(t) \dots u(t+N-1)\}$ given as

$$\min_{\mathrm{U}_t} \xi^T(t) A^T \mathbb{Q} A\xi + 2\xi^T(t) A^T B U_t + U_t^T (B^T \mathbb{Q} B + \mathbb{R}) U_t$$

In this way we are able to write a time invariant control law that stabilizes the system, which leads the robot in the origin whatever its initial position

$$\xi^T = \begin{bmatrix} x & v\cos\varphi & y & v\sin\varphi \end{bmatrix} \text{ Known}$$

$$\begin{aligned} V^N(\xi(t), u(t) \cdots u(t+N-1) &\to U_t \\ = \{u(t) \cdots u(t+N-1)\} &\to u(t|t) \end{aligned} = \begin{bmatrix} a_x \\ a_y \end{bmatrix}$$

V. SIMULATION STUDY FOR OBSTACLE AVOIDANCE

Let's consider the problem of obstacle avoidance using proposed controller that has the capacity to produce a motion without collision. To simplify the problem first consider fixed obstacles and known position. MPC algorithm chosen to control of each individual robot, the technique of artificial potential is best suited for obstacle avoidance. In this case the navigation problem is already formulated in terms of the optimization of a specific cost function, therefore it is sufficient to add a penalty to the figure of merit related to the presence of obstacles along the route. Then the following expression is obtained at the current time instant I_t is

$$\sum_{k=t}^{t+N-1}\left(\|\xi(k)\|_Q^2+\|u(k)\|_R^2\right. \\ +\sum_{p=1}^{M}\begin{cases}\dfrac{1}{\left\|(E\xi(k)-\xi_{obst_p})\right\|^2}-\dfrac{1}{d_{\lim}^2} & \left\|(E\xi(k)-\xi_{obst_p})\right\|\le d_{\lim}\\ 0 & \left\|(E\xi(k)-\xi_{obst_p})\right\|> d_{\lim}\end{cases} \tag{19}$$

Where N indicates the chosen prediction horizon, while M is the number of known obstacles along the way. The matrix E is simply a selection matrix which extract from the state vector that formed position and speed respectively along the coordinate axes, the only components related to the position to be compared with that of the obstacle.

$$\xi(k)=\begin{vmatrix}x\\ \dot{x}\\ y\\ \dot{y}\end{vmatrix}\ \xi(k)_{obst_p}=\begin{vmatrix}x_0\\ 0\\ y_0\\ 0\end{vmatrix}\rightarrow E=\begin{bmatrix}1&0&0&0\\0&0&1&0\end{bmatrix}$$

Using Eq. 19 the idea is to evaluate each state prediction along the horizon from each of the obstacles distance if this is less than the threshold value $d_{\lim}$. This approach is primarily subject to two practical problems. The first of this is the implementation of the discrete-time system. The problem in this case resides is the maximum allowable speed of movement and the chosen sampling time. Low speed and small values of the sampling period can help to alleviate the problem. The second problem which arise from the proposed approach is related to the computational complexity of the optimization process. Although the cost function used to evaluate the obstacle influence is relatively simple, a finite-dimensional real obstacle in philosophy of artificial potential need to visualize its perimeter composed of many small obstacles.

A. Numerical algorithm

Matlab environment has been chosen to test the fminunc function that allows to solve the optimization problem of non-linear functions. Using this algorithm with a prediction horizon to ensure good performance the trajectory is shown in Figure 4.

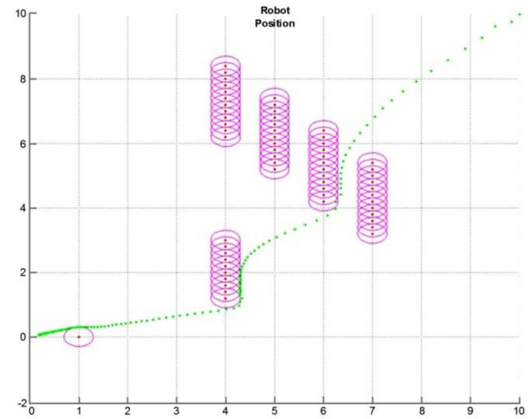

Figure 4. Obstacle Avoidance MPC using Matlab function fminunc

The numerical calculation is replaced by a simple evaluation function reducing the computational load and also the processing time. In the present case the previously obtained final expression for the sum of cost that is well suited to this type of calculation. This figure is

$$V_t^N=\|A\xi(t)+BU_t\|_Q^2+\|U_t\|_R^2+V_{obs}^N$$

With the term V_{obs}^N defined as

$$\sum_{k=0}^{N-1}\sum_{p=1}^{M}\begin{cases}\dfrac{1}{\left\|(E\xi(t+k)-\xi_{obst_p})\right\|^2}-\dfrac{1}{d_{\lim}^2} & \left\|(E\xi(t+k)-x_{obst_p})\right\|\le d_{\lim}\\ 0 & \left\|(E\xi(t+k)-x_{obst_p})\right\|> d_{\lim}\end{cases} \tag{20}$$

The penalizing component due to obstacles express the term ξ_{t+k} using the prediction in step k selecting the $k-th$ row of the block matrices A and B it can be written as

$$\xi(t+k)=A_k\xi(\mathrm{t})+B_kU_t$$

Dividing the gradient in two components respectively link to the contribution of state, input and contribution of the obstacles. These are easily achieved by remembering the quadratic form of the function selection

$$\begin{aligned}V^N&=\|A\xi(t)+BU_t\|_Q^2+\|U_t\|_R^2\\ &=(A\xi(t)+BU_t)^TQ_c(A\xi(t)+BU_t)+U^TRU\end{aligned}$$

Differentiating with respect to the entire vector of inputs U is obtained

$$\frac{\partial V}{\partial U}=2(A\xi(t)+BU_t)^TQ_cB+2U^TR$$

The component due to the obstacles having one contribution by evaluating the cost function. The cost function is then simply obtained as

$$V_2=\sum_{k=0}^{N-1}\frac{1}{\|E(A_k\xi(t)+B_kU_t-\xi_{obs})\|^2}-\frac{1}{d_{\lim}^2}$$

Its derivative with respect to the input vector U_t is

$$\frac{\partial V_2}{\partial U_t}=\sum_{k=0}^{N-1}\frac{-2}{\|E(\xi(t+k)-\xi_{obs})\|^4}(A_k\xi(t)+B_kU_t-\xi_{obs})^TE^TE\,B_k$$

Using the gradient calculated in the function fminunc, the results are still not sufficient to allow a real control of the robot even if there is a clear reduction in the calculation time. Further improvement can be made by providing the algorithms explicit form of the Hessian matrix obtained by differentiating with respect to the input vector U_t. The practical calculation of the Hessian can be done by deriving U_t with the transposed gradient.

$$\frac{\partial^2 V}{\partial U_t^2} = 2B^T QB + 2R$$

The penalizing component due to one obstacle influential for the purpose of the calculation obtained the matrix

$$\begin{aligned}\frac{\partial^2 V_2}{\partial U_t^2} = \sum_{k=0}^{N-1} \frac{8}{\|E(\xi(t+k)-\xi_{obs})\|^6} B_k^T E^T E(\xi(t+k) \\ -\xi_{obs})(\xi(t+k)-\xi_{obs})^T E^T EB_k \\ -\frac{2}{\|E(\xi(t+k)-\xi_{obs})\|^4} B_k^T E^T EB_k\end{aligned} \tag{21}$$

However, the function fminunc used so far does not allow to easily add this information to the optimization problem. For this reason optimization algorithm is chosen along with the toolbox TOMLAB.

.

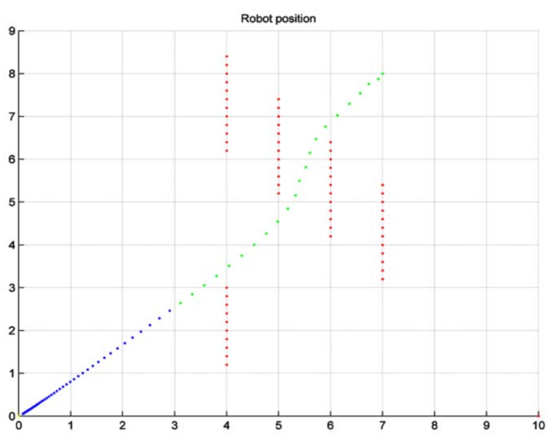

Figure 5. Obstacle avoidance using function ucSolve ()

Applying this new arrangement by using function ucsolve to the optimization problem that resolves nonlinear unconstrained problems by indicating explicitly the gradient and the Hessian. This time the result obtained associated to the minimum size of obstacles and are satisfactory for the solution and the required calculation time. With prediction horizon it ensures good performance and calculation times less than 0.2 s as shown in trajectory of Figure 5.

VI. CONCLUSION

We discussed and solved an obstacle avoidance problem while navigation of the robot towards a goal by using MPC approach that integrates with artificial potential artificial. This approach generally used for defining a proper cost function in the MPC framework. The cost function, its gradient and the Hessian in an explicit form gave us reduced computational load and the timing estimation. Such a combination provides efficient solution capable of weighing through the prediction along the horizon even the obstacles that may influence the future motion of the robot. This algorithm could be extend to test on real robots and multi-agent case. Relevant simulations demonstrate the efficacy of the proposed method.

REFERENCES

[1] D. Luders, S. Karaman, and J. P. How, "Robust sampling-based motion planning with asymptotic optimality guarantees," in *AIAA Guidance, Navigation, and Control Conference (GNC), Boston, MA*, 2013.

[2] Li, G., Yamashita, A., Asama, H., & Tamura, Y. "An efficient improved artificial potential field based regression search method for robot path planning". In *Mechatronics and Automation (ICMA),IEEE International Conference on* 2012, pp. 1227-1232)

[3] Hussein, H. Mostafa, M. Badrel-Din, O. Sultan, and A. Khamis, "Metaheuristic optimization approach to mobile robot path planning," in *1st International Conference on Engineering and Technology (ICET), New Cairo, Egypt,* 2012.

[4] Pelusi, D., Vazquez, L., Diaz, D., and Mascella, R, "Fuzzy algorithm control effectiveness on drum boiler simulated dynamics," In *Telecommunications and Signal Processing (TSP), IEEE 36th International Conference on* , 2013, pp. 272-276.

[5] Y. Gao, T. Lin, F. Borrelli, E. Tseng, and D. Hrovat, "Predictive control of autonomous ground vehicles with obstacle avoidance on slippery roads," in *Proceedings of Dynamic Systems and Control Conference (DSCC),* 2010.

[6] M. A. Abbas, J. M. Eklund, and R. Milman, *"Real-time analysis for nonlinear model predictive control of autonomous vehicles," in 25th IEEE Canadian Conference on Electrical & Computer Engineering (CCECE),* 2012, pp. 1-4.

[7] L. Grune and J. Pannek, in *Nonlinear Model Predictive Control: Theory and Algorithms.* Springer London, 2011, pp. 43-66

[8] S. J. Anderson, S. C. Peters, J. Overholt, and K. D. Iagnemma, "Semiautonomous stability control and hazard avoidance for manned and unmanned ground vehicles," in *Proceedings of the 27th Army Science Conference, Orlando,* 2010.

[9] David Mayne et al. "Robust output feedback model predictive control of contrained linear systems". *Automatica*, vol. 42, pp. 1217– 1222, 2006.

[10] Riccardo Scattolini. "Architectures for distributed and hierarchical Model Predictive Control - A review". *Journal of Process Control,* vol. 19, no. 5, pp. 723 –731, 2009.

[11] R. Fierro et al. "Hybrid Control of Formation of Robots". In: *International Conference on Robotics and Automation,* 2001, pp. 157–162.

[12] B. Lee and L. Markus, "Foundations of Optimal Control Theory." *Wiley,* New York, 1967.

[13] S. V. Rakovic´, E. C. Kerrigan, K. I. Kouramas, and D. Q. Mayne. "Invariant Approximations of the Minimal Robust Positively Invariant Set". *IEEE Transactions on automatic control,* vol.50, no.3, Mar 2005.

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