

OBSTACLE AVOIDANCE AND CONTROL OF SWEDISH WHEELED MOBILE ROBOT

Mhuammad ADIL, AhsanALI, Muhammad A. CHOUDHRY

University Of Engineering and Technology Taxila, Pakistan

Email: engr.adilfarooqi@gmail.com

Abstract: The main focus of this work is on obstacle avoidance and trajectory tracking of Swedish wheeled omnidirectional mobile robot. The proposed obstacle avoidance algorithm takes information from its onboard sensors and modulates trajectory such that the robot moves towards the goal, having maximum distance from the obstacles and minimum deviation from the trajectory. The proposed trajectory tracking controller is formulated based on linear model predictive control (MPC). The motivation of the use of linear MPC is its less computational cost relative to the nonlinear MPC. Realistic simulations are performed to test the validity and performance of the proposed obstacle avoidance algorithm and the trajectory tracking controller.

Key words: Trajectory Tracking Linear Model Predictive Control, Obstacle Avoidance, Swedish wheeled omnidirectional WMR.

1. Introduction

A standard wheel with small rollers mounted on its perimeter is called Swedish wheel. It has an extra degree of freedom to the fixed standard wheel. It was invented by Bengt Ilon in 1973 [1]. The angle γ between the rollers axis and the wheel hub axis direction is typically equal to 0° or 45° . The wheel with $\gamma = 90^\circ$ is not used, because it does not provide extra degree of freedom to the wheel. Fig. 1 is the picture of the Swedish wheel ($\gamma = 0^\circ$). Fig. 2. is the picture of 3WD omnidirectional mobile robot (a creation of IdMind - Engenharia de Sistemas, Lda) with three Swedish wheels, each wheel is 120° apart from the other. This configuration makes the robot



Figure 1: Swedish Wheel



Figure 2: Swedish Wheeled omnidirectional WMR

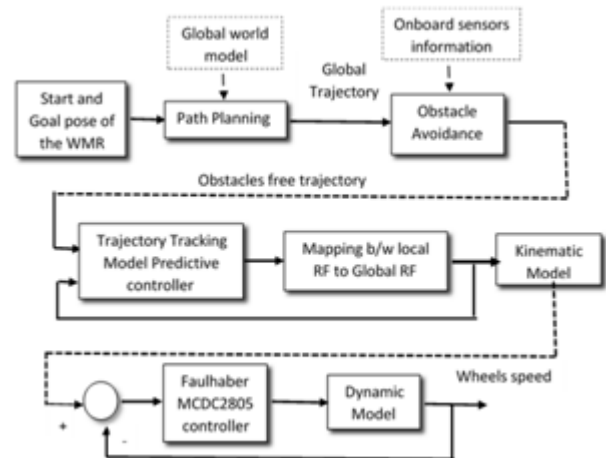


Figure 1: Navigation Architecture of WMR

truly omnidirectional, i.e. it can move in any direction without reorienting itself. The robot has 16 sonars for obstacles detection and a dioptric vision system for localization. The architecture of this robot is based on a central processing unit, a notebook PC.

This unit gathers information from all other sub-systems and sensors. The robot uses three Maxon DC 15V 90W motors for locomotion, with a 21:1 gear relation and digital encoders. Each motor is controlled by a Faulhaber MCDC2805 controller. The MCDC2805 receives speed, acceleration and position commands through its RS232 port, and interfaces directly with the motor and encoders. To control the motors by USB, there is an electronics board on the robot that translates USB signals from the PC to a standard RS232 serial signal to the controller.

Fig. 3 is the overall navigation architecture of WMR that is followed in this research work. In the path planning step robot assumes global knowledge of the environment to decide what to do over the long term to achieve its goal [2]. Road Map [3, 4], cell decomposition [5], potential field [6] are some of the strategies used for path planning. Obstacle avoidance focuses on modulating the robot's trajectory as informed by its onboard sensors during its motion, so that collision with obstacles is avoided. Bug algorithm is the simplest obstacle avoidance technique for WMRs [7]. The bug algorithm guarantees completeness, but trajectory generated by bug algorithm is very inefficient. According to the bug algorithm the robot has to fully encircle the obstacle first, then it departs from the point with the minimum distance towards the goal. In this paper we proposed an obstacle avoidance algorithm that generates shortest trajectory with minimum control effort. The proposed algorithm also contains tuning parameters that can be used for performance optimization.

The trajectory tracking control problem consists in the stabilization of error e (with respect to the position of moving reference robot) to zero. In the open loop control, the robot is not able to automatically correct or adapt the trajectory if a dynamic change of the environment occurs. So feedback controller is more appropriate approach in motion control of a mobile robot. Proportional control law presented in [8, 9] can be used to design trajectory tracking controller for omnidirectional WMR. The task of proportional control law is to find out matrix K , if it exists:

$$K = \begin{bmatrix} k_{11} & k_{12} & k_{13} \\ k_{21} & k_{22} & k_{23} \\ k_{31} & k_{32} & k_{33} \end{bmatrix}$$

With $k_{ij} = k(t, e)$ such that the control input: $\begin{bmatrix} u \\ v \\ \omega \end{bmatrix}^R = Ke$ derives error e to zero: $\lim_{t \rightarrow \infty} e(t) = 0$

The desired reference velocity or tracking error can always happen to be large enough for the actuator to reach their limits. And the Proportional control law cannot easily handle actuator constraints. We have proposed linear Model Predictive Control (LMPC) for trajectory tracking, which can efficiently handle actuator constraints. The general design objective of model predictive control is to compute a trajectory of a future manipulated variable U to optimize the future behavior of the plant output y . The optimization is performed within a limited time window by giving plant information at the start of the time window. Low level control is designed in a decentralized fashion. Dynamic coupling among the actuators are neglected, and each motor is controller separately, with a velocity PID loop to follow the speed command from inverse kinematics (Faulhaber MCDC2805 controller).

Rest of paper is divided into four sections. In section 2, mathematical model of the omnidirectional WMR is developed and validated. In section 3, proposed new obstacle avoidance algorithm, is presented. In section 4, the developed trajectory tracking controller is written. And finally, the simulation results and conclusion is written in section 5 and 6 respectively.

TABLE I
NOMENCLATURE

Symbol	DESCRIPTION	Symbol	Description
$X_I = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix}^I$	Robot's pose in global reference frame (RF)	$\begin{bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \end{bmatrix}$	Wheels angular velocity
$\dot{X}_I = \begin{bmatrix} u \\ v \\ \omega \end{bmatrix}^I$	Linear and angular velocities, in global RF	$\begin{bmatrix} r_1 \\ r_2 \\ r_3 \end{bmatrix}$	Radius of each wheel
$X_R = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix}^R$	Pose in local RF	$\begin{bmatrix} l_1 \\ l_2 \\ l_3 \end{bmatrix}$	Distance between wheels and center of the robot
$\dot{X}_R = \begin{bmatrix} u \\ v \\ \omega \end{bmatrix}^R$	Linear and angular velocities, in local RF	$\begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix}$	Angle of each wheel

2. Mathematical Modeling

Kinematic analysis of omnidirectional WMR has been addressed in several papers[10-12]. Dynamic model of omnidirectional WMR has also been developed in [13], but dynamic model of the omnidirectional WMR is not very common because of the difficulty in modeling the several internal frictions inside the wheel. Kinematic model discussed below is very similar to the one proposed by Giovanni Indeveri [14]

2.1. Kinematic Model 1: $\dot{X}_R \rightarrow \dot{X}_I$

Mapping of robot's motion in global RF in terms of local RF can be achieved by coordinate transformation [2](fig. 4).

$$\begin{bmatrix} u \\ v \\ \omega \end{bmatrix}^I = R(\theta)^{-1} \begin{bmatrix} u \\ v \\ \omega \end{bmatrix}^R$$

where $R(\theta)^{-1} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}$

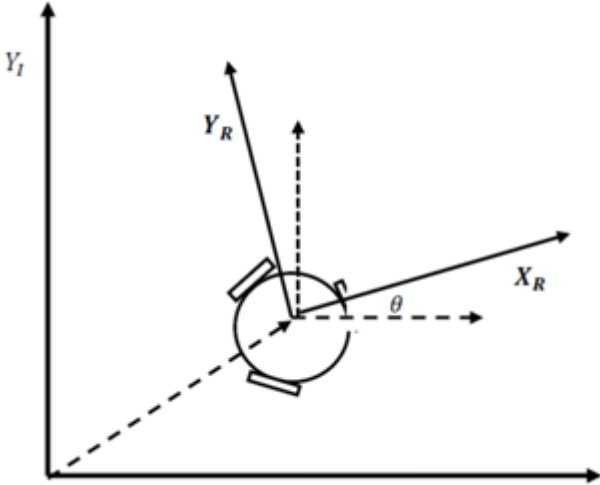


Figure 2: Coordinates transformation

2.2. Kinematic Model 2: $\dot{X}_R \rightarrow \phi_i$

Kinematic model of the WMR is a relationship between the robot speed \dot{X}_R , the wheels speed ϕ_i , and the configuration coordinates (geometric parameters of a robot). The kinematic model developed is for general setting of N number of Swedish wheels with arbitrary (but fixed) roller wheel angle can be written as [15].

$$M \begin{pmatrix} u \\ v \\ \omega \end{pmatrix} = R \omega \cos(\gamma)$$

Where u and v are the platform center velocities in the direction of x-axis and y-axis respectively, ω is the angular velocity and M is a matrix composed of $n_{\omega i}$ (unit vector parallel with roller axis) and $u_{\omega i}$ (tangent vector direction of the roller).

$$M = - \begin{bmatrix} n_{\omega 1x} & n_{\omega 1y} & r_1^T u_{\omega 1} \\ n_{\omega 2x} & n_{\omega 2y} & r_2^T u_{\omega 2} \\ \vdots & \vdots & \vdots \\ n_{\omega Nx} & n_{\omega Ny} & r_N^T u_{\omega N} \end{bmatrix}$$

For controllability $rank(M)$ should be equal to 3, and $\cos(\gamma)$ should not be equal to 0. In case of 3WD robot we have $\gamma = 0$, and each wheels are 120° apart from the other. So we can write:

$$\begin{pmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \end{pmatrix} = \frac{1}{R} \begin{pmatrix} .5 & -\frac{\sqrt{3}}{2} & -l \\ .5 & \frac{\sqrt{3}}{2} & -l \\ -1 & 0 & -l \end{pmatrix} \begin{pmatrix} u \\ v \\ \omega \end{pmatrix}^R$$

Where R is the radius of wheel and l is the distance between the center of the robot and wheel.

2.3. Empirical Model Validation

To verify the model, the kinematic model 2 is used to demonstrate whether the robot moves as expected. An overhead camera is used to observe the motion of 3WD robot. Fig. 5,6 and 7 shows the output of model (blue line) and the path followed by the 3WD omnidirectional WMR (red line) for the ramp, sinusoidal, and circular trajectories respectively. From the figures it is obvious that the error between model and the robot's output is negligible. To demonstrate the model validation results more quantitatively we have repeated the experiment for different velocities and calculated mean square error (between 3WD robot's output and the developed mathematical model output). Table II is the summary of model validation experiments.



Figure 3: Ramp Trajectory

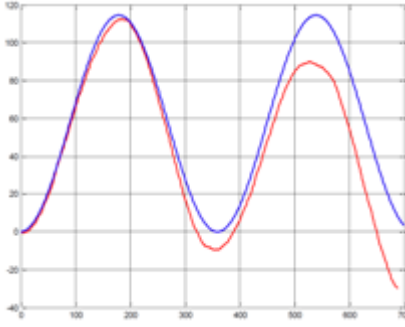


Figure 4: Sinusoidal Trajectory

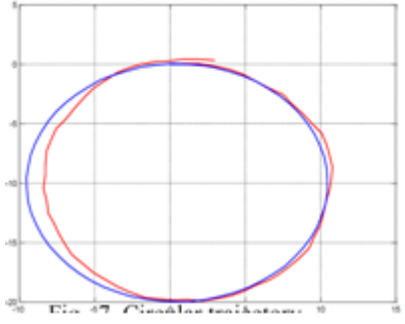


Figure 5: Circular Trajectory

TABLE II
MODEL VALIDATION

Magnitude of velocity	Mean Square Error
0.2	0.5153
0.6	0.6342
1	0.6081
1.4	0.7540
1.8	0.8437
2.2	0.9612
2.6	1.201
3	1.511
2.4	1.932
3.8	2.303

3. Obstacle Avoidance

A new method of obstacle avoidance is proposed which consists of three steps. These three steps are repeated at each sampling instant, until the robot reaches its destination. In the first step local environment of the robot is scanned, in the second step a local map is created which contains information about obstacle and goal locations, and in the third step desired new position of the robot for the next sampling instant is calculated. The calculated new position is the most optimal in the sense that it has maximum distance from the obstacles, minimum distance from the goal point, and change in control input is also minimum. Like the most of other obstacle avoidance techniques, we have considered the robot as a point capable of holonomic motion. Ultrasonic sensors are used for scanning local environment of the robot. The sensors have range between 12 cm to 5 m and accuracy of about 2 cm [16]. Ultrasonic sensors inherently suffer from several drawbacks, namely bandwidth and cross-sensitivity. But we improved the bandwidth by using a ring of ultrasonic sensors. To incorporate inaccuracies of sensors data obstacles are represented in a probabilistic fashion. A grid type model of robot's local environment is developed, in which each cell in the grid has certainty value (CV)[16]. CV is the measure of confidence that an obstacle exists within the cell area. Cells having CV less than threshold are ignored and assumed to be free of obstacles. Certainty grid is developed using same method that is proposed in well-known research work, titled "virtual field histogram"[17]. Once the certainty grid has been constructed, the obstacle avoidance algorithm needs to move the robot in such a way that obstacles are avoided and the robot proceed towards the target. The robot's linear velocity magnitude can be set at the beginning of run. And direction of the velocity is chosen such that it minimizes the objective function given by:

$$J = \Delta U^T Q \Delta U + d_{goal}^T P d_{goal} + \sum_{i=0}^N \frac{1}{P_i d_{obs_i}}$$

Where P_i is certainty value of obstacle i , d_{goal} is distance from the goal, d_{obs_i} is distance from the obstacle i , and ΔU is the change in control input. Change in control input is calculated using kinematic model 1 of the WMR. First term in the cost function penalizes the change in control input ΔU while the second term penalizes the distance from the goal.

Third term is used to maximize the distance of a robot from the obstacles. P and Q are weighting matrices that are used for tuning purpose. In the fig. 8, dotted circle $((x-x_I)^2 + (y-y_I)^2 = R)$ shows possible next locations of the robot for next sampling instant. Where $R = \text{magnitude of velocity} \times \text{sampling time}$.

Objective function is evaluated at 360 points on the circle, and the point at which the objective function gives minimum value is chosen to be the desired location of robot for the next sampling instant. To test the performance of proposed obstacle avoidance algorithm simulations were made using MATLAB with the assumption that we have perfect position information i.e. sensors or actuators noise is neglected. Some of the simulation results are given below (Fig. 9.), which shows feasibility of the proposed technique. Tuning parameters Q and P can be used to further optimize the proposed algorithm to obtain the desired behavior of robot.

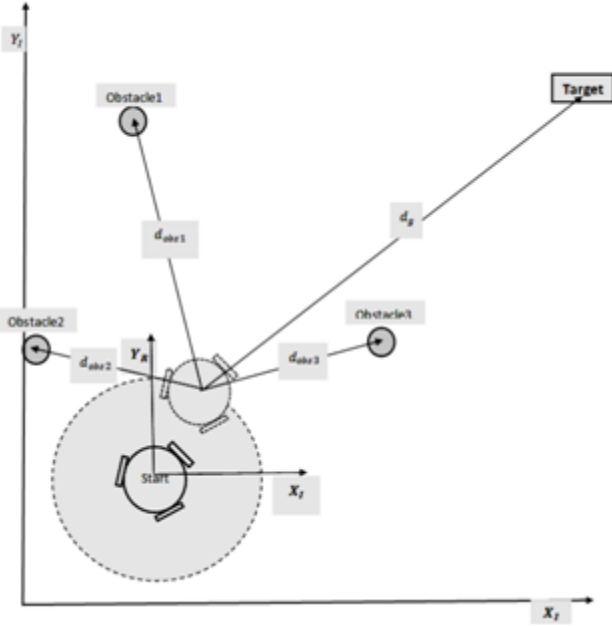


Figure 6: Obstacle Avoidance

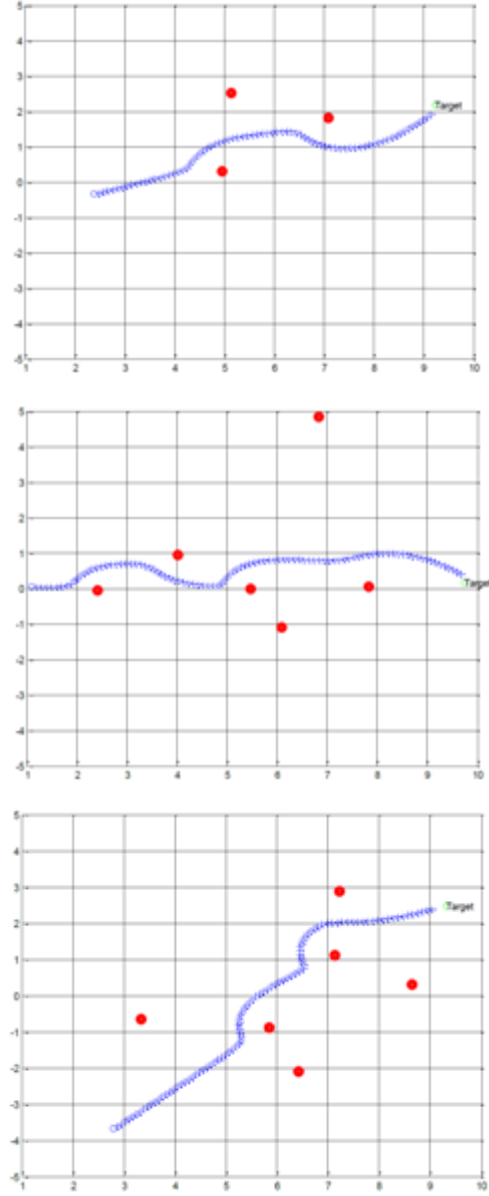


Figure 7: Simulation Results

4. Trajectory Tracking

In this section we have proposed a linear model predictive control approach for trajectory tracking of omnidirectional WMR. Although nonlinear model predictive control approach for trajectory tracking of WMR have already been proposed in literature [18], but nonlinear MPC have much more computational load than the linear MPC approach, so linear MPC is more suitable for fast moving robots. The proposed technique is similar to that proposed in [19] for trajectory tracking of nonholonomic WMR but this technique has not been used for trajectory tracking of holonomic (omnidirectional) WMR (according to the

author's best knowledge). First step of the proposed Linear MPC trajectory tracking controller is to find out a linear time varying description of the system. Then, taking control input as a decision variable, the optimization problem to be solved at each sampling instant is transformed into QP problem. Now QP problem could be solved by numerically fast and robust algorithms which lead to global optimal solutions.

Kinematic Model 1, which is simply the transformation of robot's motion form local RF into the global RF is successively linearized at the trajectory points X_{ref} and Y_{ref} .

$$X_{ref} = \begin{bmatrix} x_{vr} \\ y_{vr} \\ \theta_{vr} \end{bmatrix} \text{ and } U_{ref} = \begin{bmatrix} u_{vr} \\ v_{vr} \\ \omega_{vr} \end{bmatrix}$$

It can be supposed that given trajectory is generated by a virtual robot, which has same model as the omnidirectional WMR. We can write the linearized dynamic model of the robot as below:

$$\dot{x} = f(X_{vr}, U_{vr}) + A(X - X_{vr}) + B(U - U_{vr})$$

$$A = \left. \frac{\partial f(X, U)}{\partial X} \right|_{X = X_{vr}, U = U_{vr}} = \begin{bmatrix} 0 & 0 & -u_{vr} \sin(\theta_{vr}) - v_{vr} \cos(\theta_{vr}) \\ 0 & 0 & u_{vr} \cos(\theta_{vr}) - v_{vr} \sin(\theta_{vr}) \\ 0 & 0 & 0 \end{bmatrix}$$

$$B = \left. \frac{\partial f(X, U)}{\partial U} \right|_{X = X_{vr}, U = U_{vr}} = \begin{bmatrix} \cos(\theta_{vr}) & -\sin(\theta_{vr}) & 0 \\ \sin(\theta_{vr}) & \cos(\theta_{vr}) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Error dynamics can be written as:

$$\dot{e} = Ae + B\tilde{U}$$

Where $\dot{e} = \dot{X} - \dot{X}_{vr}$ and $\tilde{U} = U - U_{vr}$. Given the sampling time, we can obtain discrete time model of the robot by using forward difference approximation.

$$A(k) = \begin{bmatrix} 1 & 0 & u_{vr}(k) \sin(\theta_{vr}(k)) \tau - v_{vr}(k) \cos(\theta_{vr}(k)) \tau \\ 0 & 1 & u_{vr}(k) \cos(\theta_{vr}(k)) \tau + v_{vr}(k) \sin(\theta_{vr}(k)) \tau \\ 0 & 0 & 1 \end{bmatrix}$$

$$B(k) = \begin{bmatrix} \cos(\theta_{vr}(k)) \tau & -\sin(\theta_{vr}(k)) \tau & 0 \\ \sin(\theta_{vr}(k)) \tau & \cos(\theta_{vr}(k)) \tau & 0 \\ 0 & 0 & \tau \end{bmatrix}$$

Let $\bar{X}(k)$, $\bar{U}(k)$, $F(k)$ and $G(k)$ as given below.

$$\bar{X}(k+1) = \begin{bmatrix} \tilde{X}(k+1|k) \\ \tilde{X}(k+2|k) \\ \vdots \\ \tilde{X}(k+N_p|k) \end{bmatrix}$$

$$\bar{U}(k+1) = \begin{bmatrix} \tilde{U}(k|k) \\ \tilde{U}(k+1|k) \\ \vdots \\ \tilde{U}(k+N_p-1|k) \end{bmatrix}$$

$$\bar{F}(k) = \begin{bmatrix} A(k|k) \\ A(k+1|k)A(k|k) \\ A(k+2|k)A(k+1|k)A(k|k) \\ \vdots \\ A(k+N_p-1|k) \dots A(k+1|k)A(k|k) \end{bmatrix}$$

$$G(k)_1 = \begin{bmatrix} B(k|k) \\ A(k+1|k)B(k|k) \\ A(k+2|k)A(k+1|k)B(k|k) \\ \vdots \\ A(k+N_p-1|k) \dots A(k+1|k)B(k|k) \end{bmatrix}$$

$$G(k)_2 = \begin{bmatrix} 0 \\ B(k+1|k) \\ A(k+2|k)B(k+1|k) \\ \vdots \\ A(k+N_p-1|k) \dots A(k+2|k)B(k+1|k) \end{bmatrix}$$

$$G(k)_{NC} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ B(k+N_p-1|k) \end{bmatrix}$$

$$G(k) = [G(k)_1 \quad G(k)_2 \quad \dots \quad G(k)_{NC}]$$

We can write the objective function as:

$$J(k) = \bar{X}^T(k+1)Q\bar{X}(k+1) + \bar{U}^T(k)R\bar{U}(k)$$

Where Q, and R are the weighting matrices. Now we can write $\bar{X}(k+1)$ as below [21]:

$$\bar{X}(k+1) = F(k)\bar{X}(k) + G(k)\bar{U}(k)$$

Now we can write the objective function in standard QP from.

$$\bar{J}(k) = \frac{1}{2} \bar{U}^T(k) H(k) \bar{U}(k) + f^T(k) \bar{U}(k)$$

With

$$H(k) = 2(G(k)^T \bar{Q} G(k) + R)$$

$$f(k) = 2G(k)^T Q F(k) \tilde{X}(k|k)$$

Following optimization problem is solved at each sampling time

$$\bar{U}' = \underset{\bar{U}}{\operatorname{argmin}} \{ \bar{J}(k) \}$$

s.a

$$D \tilde{U}(k+i|k) \leq d \text{ where } i \in [0, N_p - 1]$$

Amplitude constraints can be written in terms of decision variable as:

$$\begin{aligned} U_{min} - U_{vr}(k+i) &\leq \tilde{U}(k+i|k) \\ &\leq U_{max} - U_{vr}(k+i) \end{aligned}$$

$$D = \begin{bmatrix} I \\ -I \end{bmatrix} \text{ and } d = \begin{bmatrix} U_{max} - U_{vr} \\ U_{min} + U_{vr} \end{bmatrix}$$

Where U_{max} and U_{min} are upper and lower limits of the deviation from the reference robot's speed, respectively.

5. Simulation Results

We have performed realistic simulation of Linear Model Predictive Trajectory Tracking controller using MATLAB 2010b. The QP problem is solved online using MATLAB QP solver (quadprog). The average computation time is 3 ms. Fig 10. shows the simulation results of circular trajectory tracking. Where blue line stands for the reference trajectory and red line represents the trajectory of WMR by using linear MPC algorithm. It is clear that the robot moves toward the trajectory and continue tracking it. We have used $N_p=20$, $N_c=5$, $P=I$, $Q=I$, these parameter can be tuned according to the desired performance criteria. Fig. 11 shows that error (vertical, horizontal, and angle errors) converges to zero. Figure 12 is the control input to each wheel, control inputs are calculated using inverse kinematics (kinematic model 2). The generated control input is very much smooth and within the actuator's limits.

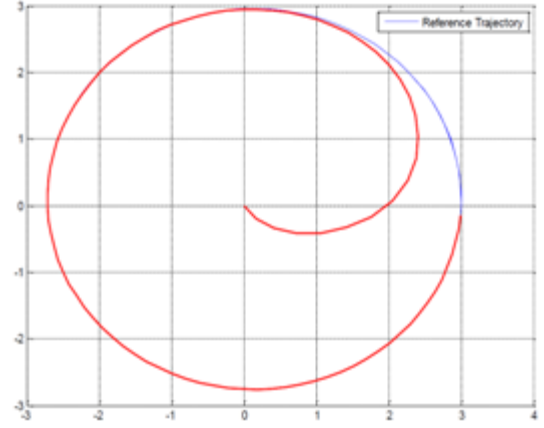


Figure 8: Circular Trajectory

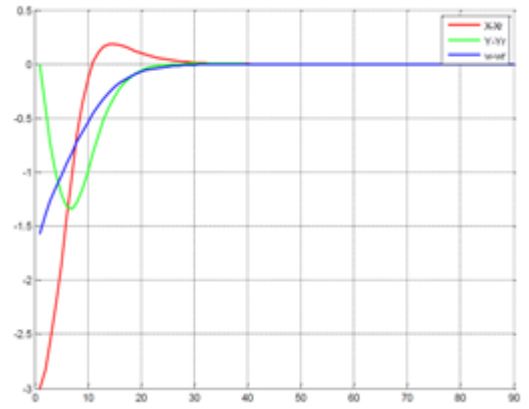


Figure 9: Error Dynamics

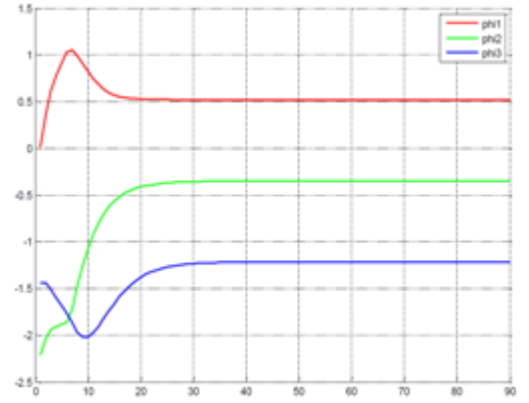


Figure 10: Control Inputs

6. Conclusion

In this paper an obstacle avoidance algorithm and trajectory tracking controller law for omnidirectional WMR is presented. A nonlinear model of the omnidirectional WMR is developed and validated experimentally, then linear time varying MPC is formulated for the trajectory tracking of the robot. In

the proposed controller optimization problem to be solved is transformed in QP problem, which can be solved using numerically fast and robust algorithm. Both trajectory tracking and obstacle avoidance algorithm are verified by performing simulations. The simulation results showed the flexibility and effectiveness of the proposed techniques.

7. References

1. Diegel, Olaf, et al. "Improved mecanum wheel design for omni-directional robots." *Proc. 2002 Australasian Conference on Robotics and Automation, Auckland*. 2002.
2. Siegwart, Roland, Illah Reza Nourbakhsh, and Davide Scaramuzza. *Introduction to autonomous mobile robots*. MIT press, 2011.
3. Kalmár-Nagy, Tamás, Raffaello D'Andrea, and Pritam Ganguly. "Near-optimal dynamic trajectory generation and control of an omnidirectional vehicle." *Robotics and Autonomous Systems* 46.1 (2004): 47-64.
4. Wu, Jianhua. *Dynamic path planning of an omni-directional robot in a dynamic environment*. Diss. Ohio University, 2005.
5. Lingelbach, Frank. "Path planning using probabilistic cell decomposition." *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*. Vol. 1. IEEE, 2004.
6. Barraquand, Jerome, Bruno Langlois, and J-C. Latombe. "Numerical potential field techniques for robot path planning." *Systems, Man and Cybernetics, IEEE Transactions on* 22.2 (1992): 224-241.
7. Ng, James, and Thomas Bräunl. "Performance comparison of bug navigation algorithms." *Journal of Intelligent and Robotic Systems* 50.1 (2007): 73-84.
8. Astolfi, Alessandro. "Exponential stabilization of a wheeled mobile robot via discontinuous control." *Journal of dynamic systems, measurement, and control* 121.1 (1999): 121-126.
9. Lee, Sung-On, et al. "A stable target-tracking control for unicycle mobile robots." *Intelligent Robots and Systems, 2000.(IROS 2000). Proceedings. 2000 IEEE/RSJ International Conference on*. Vol. 3. IEEE, 2000.
10. Samani, A. H., et al. "Design and development of a comprehensive omni directional soccer player robot." *International Journal of Advanced Robotic Systems* 1.3 (2004): 191-200.
11. Muir, Patrick F., and Charles P. Neuman. "Kinematic modeling for feedback control of an omnidirectional wheeled mobile robot." *Autonomous robot vehicles*. Springer New York, 1990. 25-31.
12. Song, Jae-Bok, and Kyung-Seok Byun. "Design and Control of a Four-Wheeled Omnidirectional Mobile Robot with Steerable Omnidirectional Wheels." *Journal of Robotic Systems* 21.4 (2004): 193-208.
13. Liu, Yong, et al. "Omni-directional mobile robot controller based on trajectory linearization." *Robotics and Autonomous Systems* 56.5 (2008): 461-479.
14. Indiveri, Giovanni. "Swedish wheeled omnidirectional mobile robots: kinematics analysis and control." *IEEE transactions on robotics* 25.1 (2009): 164-171.
15. Everett, H. R. *Sensors for mobile robots: theory and application*. AK Peters, Ltd., 1995.
16. Borenstein, Johann, and Yoram Koren. "The vector field histogram-fast obstacle avoidance for mobile robots." *Robotics and Automation, IEEE Transactions on* 7.3 (1991): 278-288.
17. Kuhne, F., Walter Fetter Lages, and J. M. Gomes da Silva Jr. "Point stabilization of mobile robots with nonlinear model predictive control." *Mechatronics and Automation, 2005 IEEE International Conference*. Vol. 3. IEEE, 2005.
18. Kühne, F., J. Gomes, and W. Fetter. "Mobile robot trajectory tracking using model predictive control." *II IEEE latin-american robotics symposium*. 2005.
19. Raffo, Guilherme V., et al. "A predictive controller for autonomous vehicle path tracking." *Intelligent Transportation Systems, IEEE Transactions on* 10.1 (2009): 92-102.