Control of Mobile Robots for Collision Avoidance

Naseem Alsadi
Department of Mechanical Engineering
McMaster University
Hamilton, Ontario, Canada
alsadin@mcmaster.ca

Gary Bone
Department of Mechanical Engineering
McMaster University
Hamilton, Ontario, Canada
gary@mcmaster.ca

S. Andrew Gadsden
Department of Mechanical Engineering
McMaster University
Hamilton, Ontario, Canada
gadsden@mcmaster.ca

Mohammad AlShabi
Department of Mechanical and Nuclear Engineering
University of Sharjah
Sharjah, United Arab Emirates
malshabi@sharjah.ac.ae

2023 Advances in Science and Engineering Technology International Conferences (ASET) | 978-1-6654-5474-2/23/$31.00 ©2023 IEEE | DOI: 10.1109/ASET56582.2023.10180627

***Abstract*—From fully autonomous warehouses to farms lacking labour; the presence of mobile robots has increased significantly. These mobile robots are commonly found in the industrial sector, where they are typically used in automating tasks. While traversing through work environments, these mobile robots must be capable of avoiding and efficiently circumnavigating both moving and stationary obstacles. To help mitigate these issues, control engineering methods are usually utilized. In this paper, we analyze two different types of mobile robots, namely holonomic and non-holonomic robots. In addition, we utilize three different control methods, namely, model predictive control (MPC), nonlinear model predictive control (NMPC), and virtual force method (VFM). Conclusively, we perform a comparative analysis of all three control methods using a variety of quantitative metrics.**

Keywords—Mobile Robots, Control Engineering, Model Predictive Control, Nonlinear Model Predictive Control

I. Introduction

The presence of mobile robots has grown rapidly in the last decade. The contemporary technological revolution has provided a relatively affordable manufacturing process for the development of robots. These robots have been utilized in various professional and industrial avenues. Examples of this are common throughout factory environments with autonomous robots, such as in Fig. 1. However, as robots become integrated throughout daily life, the chance of failure increases rapidly. Therefore, these robots must be capable of avoiding obstacles, both moving and stationary, while also arriving at their predefined destination. These robots must also be capable of dealing with sensor noise and avoiding slipping.

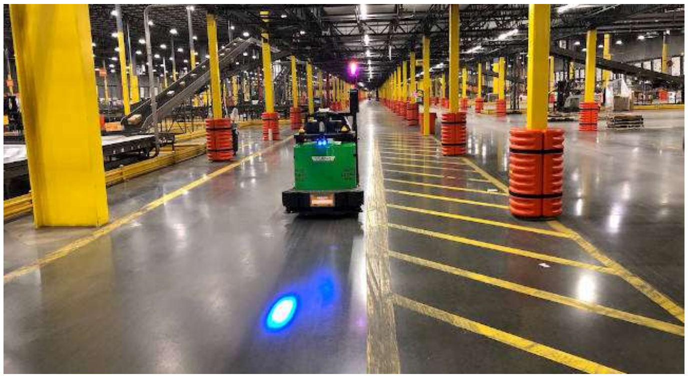

Fig. 1. Autonomous robot in a warehouse example as per [1].

The control of a mobile robot can be conducted with various control methods [2]-[5]. In this paper, we look at the control of two variant types of mobile robots. The first of which is the holonomic robot, which is also referred to as the omnidirectional robot [5], [6]. The holonomic robot is capable of rotation and cartesian translation. The second type of mobile robot we utilize is a non-holonomic robot or differential drive robot [7], [8]. This type of robot cannot translate left/right.

To control both the holonomic robot and non-holonomic robot, we employ three variant control methods, namely MPC, NMPC, and VFM. Each of these methods is explored in their individual section and later comparatively assessed against one another.

A. Differential Drive Mobile Robots

The information provided in this section summarizes the plant information utilized in this paper. The cartesian acceleration of the mobile robot is the controllable variable across all controllers. The mobile robot's location and speed are achieved by numerical integration. A differential-drive robot, like that shown in Fig. 2, uses separate electric motors to power its wheels. We assume that, in relation to the mass of the robot, the inertias of the motors and wheels are insignificant. Furthermore, we assume that the motors are speed-controlled and incapable of being reversed.

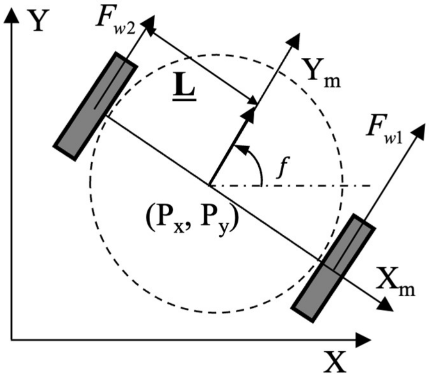

Fig. 2. Differential drive robot geometry used in this study.

We assume the ground is level, and the robot's path is either straight or moderately curved such that no sliding occurs in the Y_m direction. The relationship between the velocities in the robot's moving coordinate frame (X_m, Y_m) and the fixed coordinate frame (X, Y) is given by the following:

$$\begin{matrix}\dot{x}\\ \dot{y}\\ \dot{f}\end{matrix} = \begin{bmatrix} cosf & 0\\ sinf & 0\\ 0 & 1\end{bmatrix}\begin{bmatrix}\dot{y}_m\\ \dot{f}\end{bmatrix} \tag{1}$$

The linear velocities of the wheel centers are:

$$v_{w1} = \dot{y}_m + L\dot{\phi} \tag{2}$$

$$v_{w2} = \dot{y}_m - L\dot{\phi} \tag{3}$$

The net force and torque applied to the robot by its wheels are:

$$F_y = F_{w1} + F_{w2} \tag{4}$$

$$\tau_z = L\left(F_{w1} - F_{w2}\right) \tag{5}$$

Assuming the robot can be modelled as a uniform cylinder of mass m_r, and radius r_c, the resulting linear and angular accelerations are:

$$\ddot{y}_m = \frac{F_y}{m_r} \tag{6}$$

$$\ddot{\phi} = \frac{\tau_z}{\frac{1}{2} m_r r_c^{\,2}} \tag{7}$$

When the wheels are rolling (i.e., not slipping):

$$F_{w1} = \left(\frac{m_r r}{2}\right)\ddot{\theta}_{m1} \tag{8}$$

$$F_{w2} = \left(\frac{m_r r}{2}\right)\ddot{\theta}_{m2} \tag{9}$$

where $\ddot{\theta}_{\mathrm{m}}$ is the angular acceleration of the motor, and r is the radius of the wheels. If wheel 1 is currently rolling, then it will start slipping when:

$$\left|\ddot{\theta}_{m1}\right| > \frac{g\mu_{s1}}{r} \tag{10}$$

where μ_{s1} is the static coefficient of friction for wheel 1. Similarly, wheel 2 will start slipping when:

$$\left|\ddot{\theta}_{m2}\right| > \frac{g\mu_{s2}}{r} \tag{11}$$

If wheel 1 is slipping, then the force it applies to the robot becomes:

$$F_{w1} = \tfrac{1}{2}\mu_{k1} m_r g \operatorname{sign}\left(\dot{\theta}_{m1} r - v_{w1}\right) \tag{12}$$

where $\mu_{k1}$is its kinetic coefficient of friction and $\mu_{k1} < \mu_{s1}$. Similarly, if wheel 2 is slipping:

$$F_{w2} = \tfrac{1}{2}\mu_{k2} m_r g \operatorname{sign}\left(\dot{\theta}_{m2} r - v_{w2}\right) \tag{13}$$

If a wheel is currently slipping, then it will start rolling when (10) and (11) are false, and $v_w = \dot{\theta}_m r$. In a numerical simulation, the conditions to check for wheel 1 are:

$$\left|\ddot{\theta}_{m1}\right| \le \frac{g\mu_{s1}}{r} \tag{14}$$

$$\left|v_{w1} - \dot{\theta}_{m1} r\right| < v_{tolerance} \tag{15}$$

For wheel 2, they are:

$$\left|\ddot{\theta}_{m2}\right| \le \frac{g\mu_{s2}}{r} \tag{16}$$

$$\left|v_{w2} - \dot{\theta}_{m2} r\right| < v_{tolerance} \tag{17}$$

II. Model Predicitive Control

A. Introduction

Model predictive control (MPC) is a control method which belongs to a category of control methods which explicitly utilize a dynamic model of the plant to predict the effect of future actions on the output. The main benefit of deploying MPC is its ability to provide optimization for the current timestep while also accounting for the future timesteps simultaneously. This is in comparison to simple controllers such as the PID which do not possess this predictive capability. Optimization is fundamental to the procedure the MPC carries out [9]–[11].

MPC uses the current plant measurements, the current dynamic state of the process, the MPC models, and the process variable targets and limits to calculate future changes in the dependent variables. These changes are calculated to hold the dependent variables close to the target while honoring constraints on both independent and dependent variables. The MPC typically sends out only the first change in each independent variable to be implemented and repeats the calculation when the next change is required.

B. Methodology

The MPC procedure begins with acquiring the current state $x(t)$ from the desired plant. The state-space model describing the system can be formulated as:

$$x(t) = Ax(t-1) + Bu(t-1) \tag{18}$$

$$y(t) = Cx(t) \tag{19}$$

where u is the input vector, A is the state matrix and B is the input matrix. At timestep t the MPC will take the current state as the initial state.

$$x_0 = x(t) \tag{20}$$

Subsequently, the model will perform an online optimization procedure where the optimal input vector, $u(t)$, is calculated given the initial state, x_0. The process can be formulated with:

$$\begin{gathered} Minimize \sum_{i=0}^{K} L(x_i\,, u_i). \\ subject\ to\ x_i = Ax_i + Bu_i \end{gathered} \tag{21}$$

Multiple constraints can be further applied to ensure that the estimated optimal input and state do not exceed the predefined boundaries.

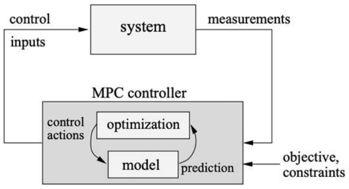

Fig. 3. Typical schematic of the MPC strategy as per [12].

Figure 3 depicts the cumulative procedure where the MPC is provided with the given system state, which can be achieved with measurement tools such as sensors. The model then performs the online optimization task given the measured state. Subsequently, the calculated optimal input vector is provided to the plant. The plant then applies the calculated input vector and therein updates the current system state. The procedure is repeated until a stopping condition is met.

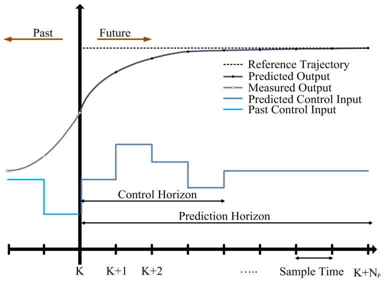

Fig. 4. Example of the MPC strategy [13].

Figure 4 shows an example of the MPC procedure over a predefined horizon of K. The optimal input to the given state is taken at the present step.

C. Holonomic Control

In this section the MPC method is applied to the control of a holonomic mobile robot. The input vector is comprised of the robot's cartesian acceleration, or it's acceleration in the x and y direction. Position and velocity are obtained with the application of numerical integration.

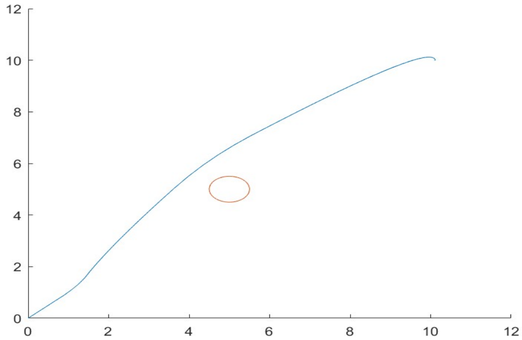

Fig. 5. MPC Holonomic: Horizon = 15, $r_{robot} = 0.5$, $r_{obstacle} = 0.5$.

To evaluate the results of the MPC method on the control of a holonomic robot we drive the robot to a destination of $P_x = 10$ and $P_y = 10$, where an obstacle is positioned at $O_x = 4$ and $O_y = 4$. Various parameters are modified to provide a cumulative depiction of the performance of the MPC, as shown through Figs. 5-7.

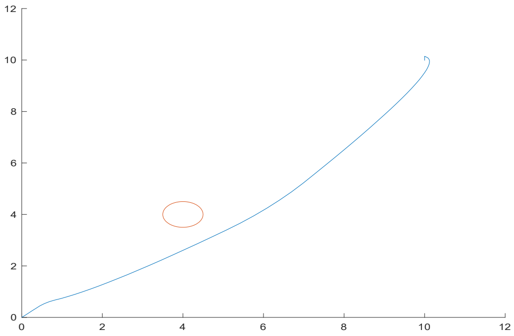

Fig. 6. MPC Holonomic: $O_x = 4$ *and* $O_y = 4$.

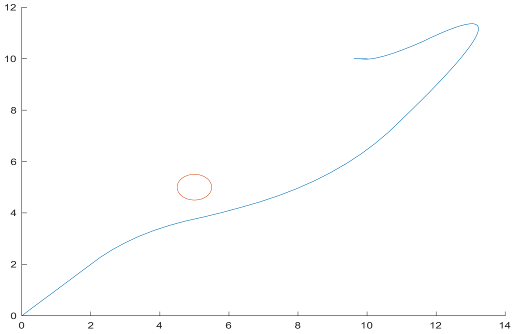

Fig. 7. MPC Holonomic: Horizon = 10.

D. Non-Holonomic Control

To evaluate the results of the MPC method on the control of a non-holonomic robot we drive the robot to a destination of $P_x = 10$ and $P_y = 10$, where an obstacle is positioned at $O_x = 2$ and $O_y = 2$. As shown in Fig. 8, various parameters are modified to provide a cumulative depiction of the performance of the MPC.

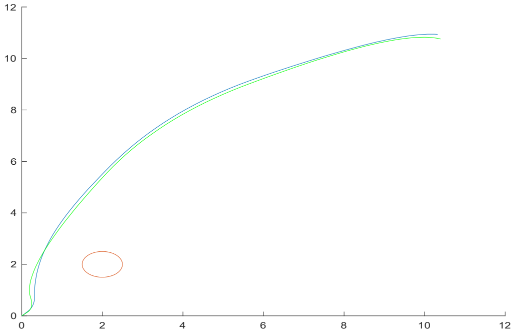

Fig. 8. MPC Non-Holonomic: $O_x = 2$ *and* $O_y = 2$.

a) Assessing Slipping

To examine the effects of slipping on the MPC controller we perform a controlled analysis of the effects of modifying the wheel friction coefficient. As shown in Fig. 9, we start with setting a symmetrical wheel friction coefficient of $\mu_{s_1} = \mu_{s_1} = 0.8$.

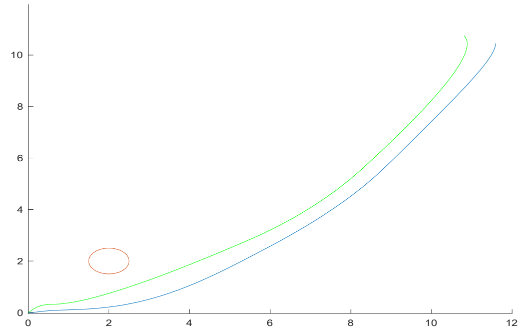

Fig. 9. MPC Non-Holonomic with $\mu_{s_1} = \mu_{s_1} = 0.8$.

When a wheel friction coefficient of $\mu_{s_1} = \mu_{s_1} = 0.8$ is introduced, we can see that the MPC struggles to follow the desired trajectory. The estimated mean squared error is 0.1520. Subsequently, a wheel friction coefficient of $\mu_{s_1} = \mu_{s_1} = 0.4$ is examined. The resultant mean squared error is 0.2302.

Sensor noise was added and its effect on the movement of the robot is summarized by the results in Table 1. The noise is added with the following equation:

$$Sensor\ Noise = Noise\ Size * 2 * (rand - 0.5)$$

Table 1: Mean Squared Error and Mean Absolute Error of Non-Holonomic MPC with Added Sensor Noise.

Noise Size	Mean Squared Error	Mean Absolute Error
0	0.1028	0.2642
0.001	0.0991	0.2595
0.005	0.1149	0.2790
0.01	0.2012	0.3731

III. Nonlinear Model Predictive Control (NMPC)

The inclusion of nonlinear system models in the prediction is a distinctive feature of nonlinear model predictive control, or NMPC, a subset of model predictive control [14]–[18].

A. Holonomic Control

In this section, the control of a holonomic robot with NMPC is assessed and the results are shown in Figs. 10 and 11. Similar to the last section, due to page constraints, we only include the most interesting figures of the results.

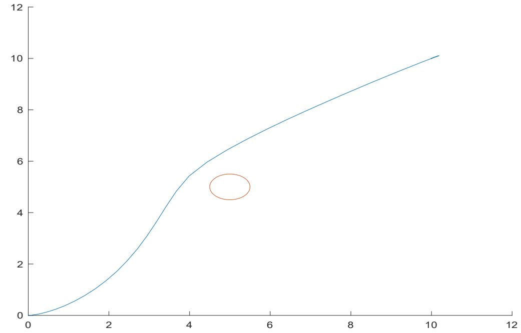

Fig. 10. NMPC Holonomic: $O_x = 5\ and\ O_y = 5$.

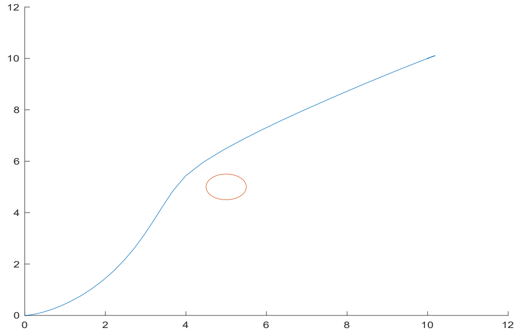

Fig. 11. NMPC Holonomic: Horizon = 40.

B. Non-Holonomic Control

In this section, the control of a non-holonomic robot is conducted with the employment of NMPC. To assess the performance of the NMPC method on the control of a non-holonomic robot we drive the robot to a destination of $P_x = 10$ and $P_y = 10$, where an obstacle is positioned at $O_x = 5$ and $O_y = 5$. These results are shown in Fig. 12.

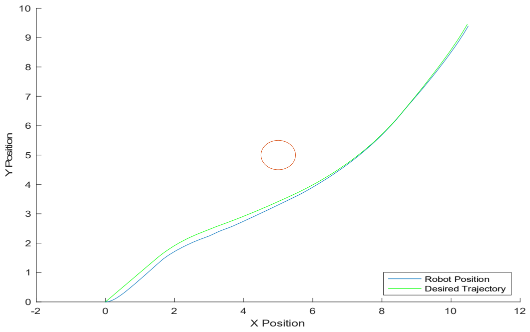

Fig. 12. NMPC Non-Holonomic: $O_x = 5\ and\ O_y = 5$.

Similar to before, to examine the effects of slipping, we start with setting a symmetrical wheel friction coefficient of $\mu_{s_1} = \mu_{s_1} = 0.8$.

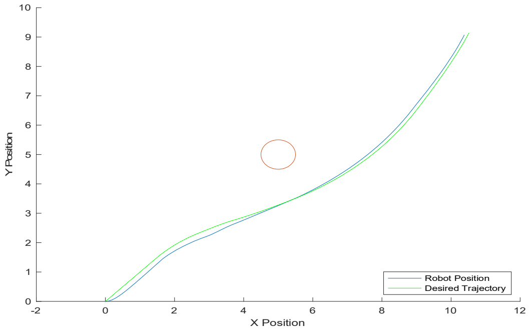

Fig. 13. NMPC Non-Holonomic: $\mu_{s_1} = \mu_{s_1} = 0.8$.

We can see that from Fig. 13 there is minimal slipping, and the robot manages to reach its destination with an estimated mean squared error of 0.0191. We subsequently reduce the coefficient of friction to a smaller value of $\mu_{s_1} = \mu_{s_1} = 0.4$. In this case, we should expect to see more slipping and therefore greater error.

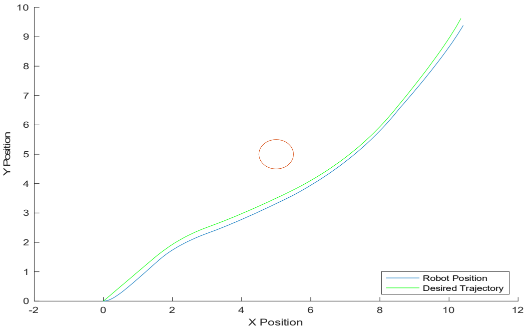

Fig. 14. NMPC Non-Holonomic: $\mu_{s_1} = \mu_{s_1} = 0.4$.

The results clearly depict that the error increased when the coefficient of friction was decreased, as shown in Fig. 14. The estimated mean squared error was 0.0283. In later sections, the NMPC is compared with variant control methods in its ability to minimize error while slipping. Table 2 summarizes the errors obtained under varying noise signals.

Table 2: Mean Squared Error and Mean Absolute Error of Non-Holonomic NMPC with Added Sensor Noise

Noise Size	Mean Squared Error	Mean Absolute Error
0	0.0167	0.1003
0.001	0.0167	0.1003
0.005	0.0168	0.1006
0.01	0.0168	0.1007
0.1	0.0207	0.1093

IV. Virtual Force Method (VFM)

The VFM is a control method which was presented for the use of obstacle-avoiding mobile robots in 2012 by Zeng and Bone et al. [3]. The method works by utilizing three fundamental virtual forces, contingent on the robot's location relative to the obstacle [19]–[21]. The method is summarized by Fig. 15.

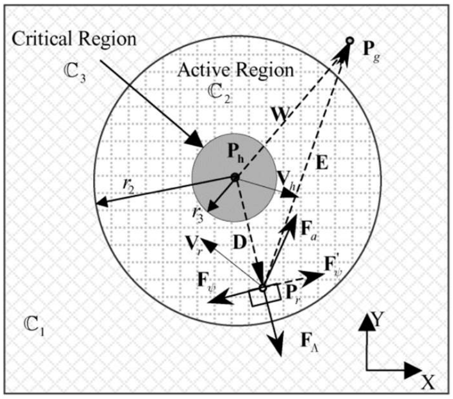

Fig. 15. VFM regions as illustrated in [3].

Two regions are defined for the calculation of the appropriate virtual forces. The first of which is the active region, C2. The active region C2 is defined as the region near the human. If any mobile robot enters the region defined as C2, both the repulsive and detour virtual forces will be engaged to cause the mobile robot to avoid the obstacle. The critical region C3 is the region closest to the obstacle, and it is very dangerous if a robot intrudes into this region. If the robot enters this area, the threat of a crash is large. Therefore, the mobile robot is decelerated to a stop.

We can calculate the respective force values by using this set of equations:

$$E = P_g - P_r \tag{22}$$

$$D = P_r - P_h \tag{23}$$

$$W = P_g - P_h \tag{24}$$

The attractive virtual force is employed to direct the mobile robot towards the destination. This force is engaged when the robot enters C1 or C2. The force can be formulated as:

$$F_a = \begin{cases} K_1 E + K_2 \dot{\mathrm{E}} & \text{if } \mathrm{P_r} \in \mathrm{C}_2 \cup \mathrm{C}_1 \\ \quad Undefined & \text{if } \mathrm{P_r} \in \mathrm{C}_3 \end{cases} \tag{25}$$

A subsequent repulsive virtual force is employed to force the robot away from the obstacle. The repulsive force is engaged when the robot enters C2.

$$d = \|D\|_2 \tag{26}$$

$$\wedge = \frac{(r_2 - d\,)^2}{d - r_3} \tag{27}$$

$$\wedge^* = -d\,\frac{(r_2 - d\,)^2}{(d - r_3)^2} \tag{28}$$

$$F_\wedge = \begin{cases} K_3 \wedge + K_4 \wedge^* & \text{if } \mathrm{P_r} \in \mathrm{C}_2 \\ Undefined & \text{if } \mathrm{P_r} \in \mathrm{C}_1 \cup \mathrm{C}_3 \end{cases} \tag{29}$$

The detour force is the last force applied to the robot and its main objective to allow the robot to detour while keeping the destination in mind.

$$F_\Phi = \begin{cases} (K_5 \Phi + K_6 \Phi^*)\boldsymbol{u}_\Phi & \text{if } \mathrm{P_r} \in \mathrm{C}_2 \\ Undefined & \text{if } \mathrm{P_r} \in \mathrm{C}_1 \cup \mathrm{C}_3 \end{cases} \tag{30}$$

The VFF is the combination of the three forces. Thus, the VFF for the robot with a single human/obstacle is:

$$F_\mathrm{v} = \begin{cases} F_a & \text{if } \mathrm{P_r} \in \mathrm{C}_1 \\ F_a + F_\wedge + F_\Phi & \text{if } \mathrm{P_r} \in \mathrm{C}_2 \\ Undefined & \text{if } \mathrm{P_r} \in \mathrm{C}_3 \end{cases} \tag{31}$$

The holonomic dynamics can be viewed as that of a point mass. Therefore, the acceleration vector can be obtained from F_v. The positional information about the reference robot can be obtained by numerical integration.

A. Holonomic Control

To evaluate the results of the VFM on the control of a holonomic robot, we drive the robot to a destination of $P_x = 10$ and $P_y = 10$, where an obstacle is positioned at $O_x = 4$ and $O_y = 4$. Various parameters are modified to provide a cumulative depiction of the performance of the MPC. These results are shown in Figs. 16 and 17.

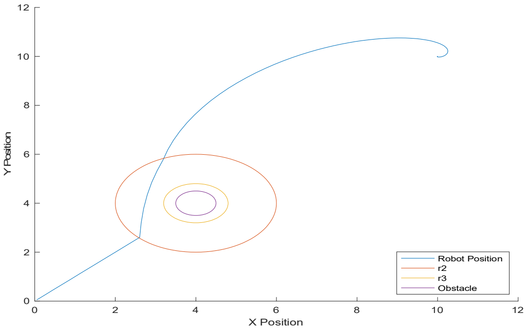

Fig. 16. VFM Holonomic: $O_x = 4 \; and \; O_y = 4$.

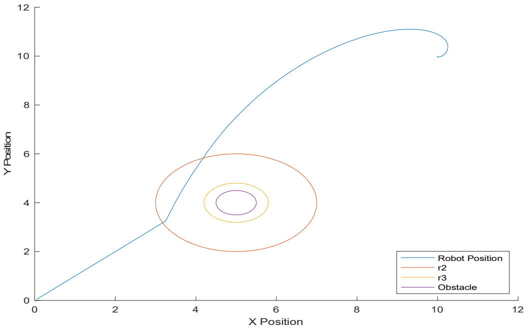

Fig. 17. VFM Holonomic: $O_x = 5 \; and \; O_y = 4$.

B. Non-Holonomic Control

In this section, the control of a non-holonomic robot is conducted with the employment of VFM. To assess the performance of the VFM on the control of a non-holonomic robot we drive the robot to a destination of $P_x = 10$ and $P_y = 10$, where an obstacle is positioned at $O_x = 5$ and $O_y = 5$. Various parameters were modified to provide a cumulative depiction of the performance of the VFM. The results are shown in Figs. 18 and 19.

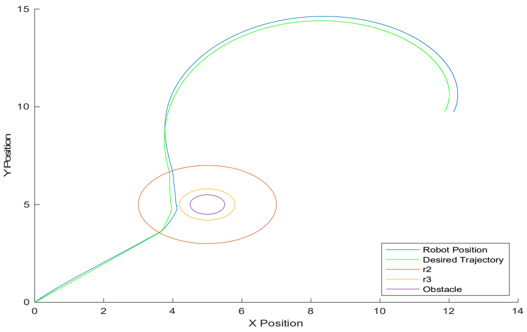

Fig. 18. VFM Non-Holonomic: $O_x = 5 \; and \; O_y = 5$.

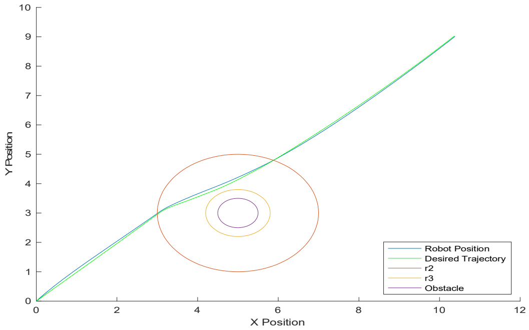

Fig. 19. VFM Non-Holonomic: $O_x = 5 \; and \; O_y = 3$.

Similar to before, to examine the effects of slipping, we start with setting a symmetrical wheel friction coefficient of $\mu_{s_1} = \mu_{s_1} = 0.8$. Note that mass was increased to 10 kg for this experiment due to terrible slipping performance by the VFM.

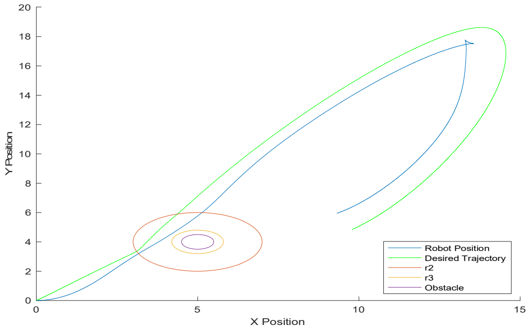

Fig. 20. VFM Non-Holonomic: $\mu_{s_1} = \mu_{s_1} = 0.8$.

Figure 20 clearly depicts that the VFM is heavily affected by slipping. The robot manages to overcome the obstacle, however, never reaches its destination. We subsequently reduce the coefficient of friction to a smaller value of $\mu_{s_1} = \mu_{s_1} = 0.4$. In this case, we should expect to see more slipping and therefore greater error

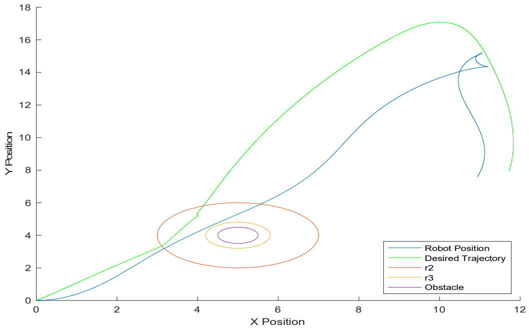

Fig. 21. VFM Non-Holonomic: $\mu_{s_1} = \mu_{s_1} = 0.4$.

Figure 21 clearly depicts that the error increased when the coefficient of friction was decreased. The estimated mean squared error was 1.4299. In later sections, the NMPC is compared with variant control methods in its ability to minimize error while slipping. Table 3 summarizes the results of changing the noise signals on the error.

Table 3: Mean Squared Error and Mean Absolute Error of Non-Holonomic VFM with Added Sensor Noise

Noise Size	Mean Squared Error	Mean Absolute Error
0	0.0021	0.0303
0.001	0.0167	0.1003
0.005	0.0031	0.0369
0.01	0.0054	0.0524
0.1	0.0061	0.0627

V. Moving Obstacles

In this section each of the controllers will be assessed with moving obstacles. The path of the obstacles is kept the same for all of the controllers to provide a cumulative and fair overview of the performance of each controller. Similar to the last section, due to page constraints, we only include the most interesting figures of the results.

Figures 22 and 23 show the results of implementing MPC, Figs. 24 and 25 show the results of implementing the NMPC, and Figs. 26 and 27 show the results of implementing VFM.

A. Model Predictive Control (MPC)

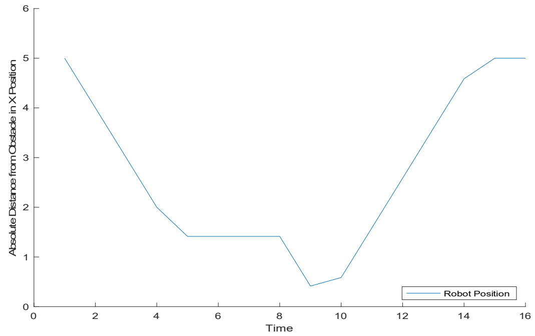

Fig. 22. Absolute Distance Away in X Position (MPC).

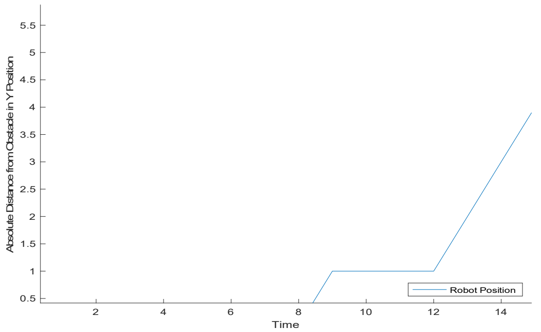

Fig. 23. Absolute Distance Away in Y Position (MPC).

B. Nonlinear Model Predictive Control (NMPC)

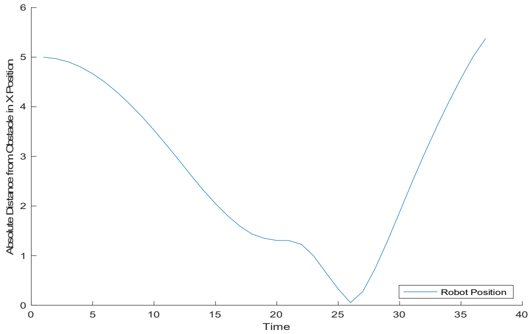

Fig. 24. Absolute Distance Away in X Position (NMPC).

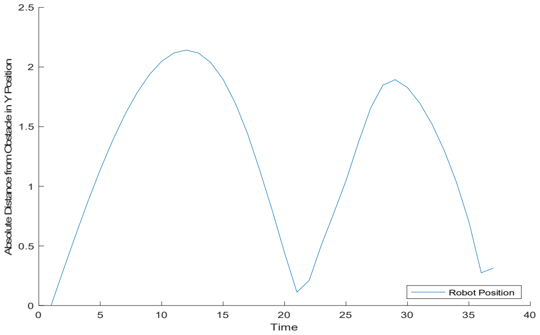

Fig. 25. Absolute Distance Away in Y Position (NMPC).

C. Virtual Force Method (VFM)

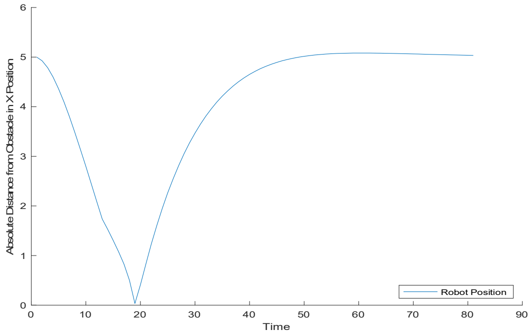

Fig. 26. Absolute Distance Away in X Position (VFM).

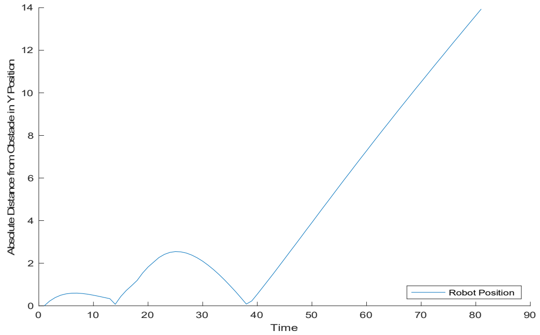

Fig. 27. Absolute Distance Away in Y Position (VFM).

VI. Comparative Assessment of Techniques

A. Overview of Metrics

To compare the performance of the three different controllers we utilize two qualitative metrics, namely mean squared error and mean absolute error, where:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(Y_i - \hat{Y}_i\right)^2 \qquad (32)$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|y_j - \hat{y}_j\right| \qquad (33)$$

In addition to these two metrics, we also evaluate the performance of the holonomic model on its ability to reach the destination while avoiding the obstacle in the shortest time frame.

B. Evaluation of Holonomic Control

1) Evaluation of Varying Horizon Values

A key parameter for the MPC and NMPC is the horizon value. This value decides how far into the future to account for. Therefore, we should expect the accuracy and efficiency of the control method to increase as this parameter increases as well. To test this the effect of modifying the horizon values on the accuracy and efficiency of the robot, we compare three different horizon values.

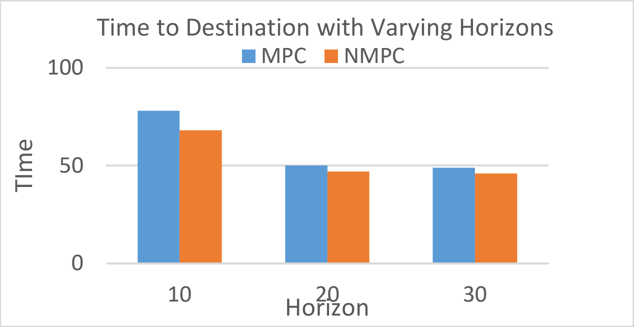

Fig. 28. Time to Destination for Variant Horizon Values.

The results shown in Fig. 28 show that as the horizon value increases, the number of timesteps to reach the destination decreases. This is what we expect because as the horizon value increases, so does the efficiency of the robot as it makes more educated decision earlier on in the moving process.

C. Evaluation of Non-Holonomic Control

1) Evaluation of Varying Friction Coefficients

All three non-holonomic models were compared in their ability to handle varying friction coefficients. Figure 29 shows the calculated mean squared error for two variant symmetrical wheel friction coefficients. The plot on depicts the error for the MPC and NMPC. The VFM method was unable to follow the desired trajectory if slipping is introduced. Therefore, only the MPC and NMPC are compared since the error for the VFM was much larger.

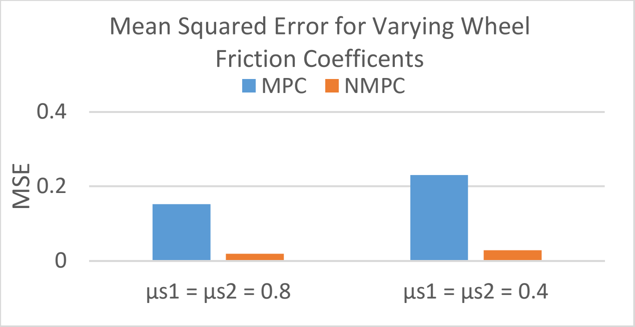

Fig. 29. MSE for Varying Wheel Friction Coefficients.

The results indicate that the NMPC was superior in maintaining the desired trajectory with varying wheel friction coefficients. This can be due to the nonlinear nature of the robot movement when slipping is introduced.

2) Evaluation of Error with Introduction of Sensor Noise

Figures 30 and 31 summarize the results of varying the amplitude of sensor noise and its effects on the controllers.

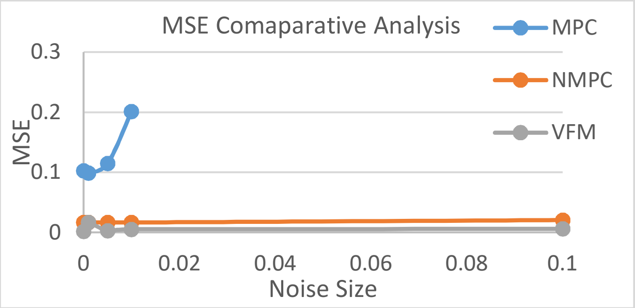

Fig. 30. MSE Comparative Analysis for MPC, NMPC, and VFM.

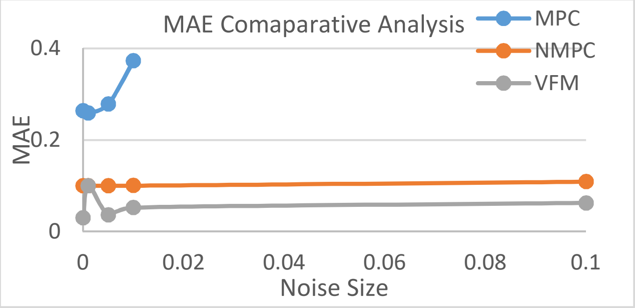

Fig. 31. MAE Comparative Analysis for MPC, NMPC, and VFM.

The results show that with added random sensor noise we experience an increase in error regardless of the metric or controller used. This is due to the controller receiving the current state measurement with added noise. Therefore, the predicted optimal input will typically be offset from its true value. In addition, the controllers are also individually assessed in their ability to handle sensor noise. The results show that sensor noise was most optimally handled with VFM and, subsequently the NMPC. The VFM had the lowest MSE and MAE across numerous experiments with varying noise size. The MPC on the other hand, had the highest MSE and MAE in all the experiments.

VII. Discussion and Concluding Remarks

In this paper, we look at the control of two variant types of mobile robots. The first of which is the holonomic robot, which is also referred to as the omnidirectional robot. The holonomic robot is capable of rotation and cartesian translation. To control the holonomic robot, we employ three variant control methods, namely MPC, NMPC, and VFM. Each of the control methods was successful in avoiding a stationary or moving obstacle while reaching a predefined destination point. The models were tested with various destination and obstacle locations.

The MPC and NMPC were able to reach the destination in the least amount of time while making the least amount of unnecessary movement around the obstacle. The NMPC was able to reach the destination in slightly less time than MPC. While the VFM was able to avoid the obstacle and reach the destination, the movement made by the robot increased the overall time to reach the destination. This can be combatted by carefully tunning the gain values in the VFM equation; however, this is not ideal in comparison to the other methods.

The holonomic robots controlled with the MPC and NMPC were further analyzed by assessing the change in output when the horizon value is modified. We find that as the horizon value is increased, the efficiency and accuracy of the model increase as well. This is due to the controller taking into account a greater number of future steps when the horizon value is increased. The second type of robot which was analyzed was the non-holonomic robot. This robot is not capable of left/right translation. The non-holonomic robot was controlled with the same three controllers. The performance of the controllers was assessed with variant experiments.

The first experiment we conducted was on the influence of slipping and the controller's ability to maintain the desired trajectory and reach its destination. We found that the NMPC performed the best in reducing error while slipping. Subsequently, the effect of sensor noise was analyzed. Modifying the size of random sensor noise, we compared the performance of each controller. Utilizing both MSE and MAE as metrics, the VFM was identified as the best performing model in achieving the lowest error regardless of the size of the sensor noise. In addition, we utilized moving obstacles to examine the performance of the controlled robots in a pragmatic environment. All three controllers were able to avoid the moving obstacle. This is an excellent test of the ability to use these controllers in an environment with human workers.

VIII. Conclusion

In summary, all three controllers succussed in performing the required task. However, the NMPC strategy was the most robust to the various situations and scenarios. In addition, it was faster than the MPC in its execution. Therefore, in this study, the NMPC strategy was found to be the ideal option for both the holonomic and non-holonomic robots within the selected testing environment and conditions.

IX. References

[1] "Control Engineering | Choosing the right robot for warehousing, manufacturing operations," *Control Engineering*, Aug. 15, 2021. https://www.controleng.com/articles/choosing-the-right-robot-for-warehousing-manufacturing-operations/ (accessed Apr. 28, 2022).

[2] G. Klančar and I. Škrjanc, "Tracking-error model-based predictive control for mobile robots in real time," *Robot. Auton. Syst.*, vol. 55, no. 6, pp. 460–469, Jun. 2007, doi: 10.1016/j.robot.2007.01.002.

[3] "Mobile Robot Navigation for Moving Obstacles with Unpredictable Direction Changes, Including Humans: Advanced Robotics: Vol 26, No 16." https://www.tandfonline.com/doi/abs/10.1080/01691864.2012.703166?journalCode=tadr20 (accessed Apr. 28, 2022).

[4] Z. Zhang, W. Jiang, and S. S. Ge, "Adaptive Tracking Control of Nonholonomic Mobile Robots With Input Constraints and Unknown Disturbance," in *2022 13th Asian Control Conference (ASCC)*, May 2022, pp. 396–402. doi: 10.23919/ASCC56756.2022.9828071.

[5] I. Doroftei, V. Grosu, and V. Spinu, "Omnidirectional Mobile Robot - Design and Implementation," in *Bioinspiration and Robotics Walking and Climbing Robots*, M. K., Ed. I-Tech Education and Publishing, 2007. doi: 10.5772/5518.

[6] K. Watanabe, "Control of an omnidirectional mobile robot," in *1998 Second International Conference. Knowledge-Based Intelligent Electronic Systems. Proceedings KES'98 (Cat. No.98EX111)*, Apr. 1998, vol. 1, pp. 51–60 vol.1. doi: 10.1109/KES.1998.725827.

[7] Y. Chung, C. Park, and F. Harashima, "A position control differential drive wheeled mobile robot," *IEEE Trans. Ind. Electron.*, vol. 48, no. 4, pp. 853–863, Aug. 2001, doi: 10.1109/41.937419.

[8] S. K. Malu and J. Majumdar, "Kinematics, Localization and Control of Differential Drive Mobile Robot," *Glob. J. Res. Eng.*, vol. 14, no. H1, pp. 1–7, Jan. 2014.

[9] E. F. Camacho and C. B. Alba, *Model Predictive Control*. Springer Science & Business Media, 2013.

[10] B. Kouvaritakis and M. Cannon, *Model Predictive Control*. Cham: Springer International Publishing, 2016. doi: 10.1007/978-3-319-24853-0.

[11] J. B. Rawlings, "Tutorial overview of model predictive control," *IEEE Control Syst. Mag.*, vol. 20, no. 3, pp. 38–52, Jun. 2000, doi: 10.1109/37.845037.

[12] M. Arnold, R. R. Negenborn, G. Andersson, and B. De Schutter, "Multi-Area Predictive Control for Combined Electricity and Natural Gas Systems," *2009 Eur. Control Conf. ECC 2009*, Mar. 2015.

[13] "Model Predictive Control: An Overview and Selected Applications | AIChE." https://www.aiche.org/academy/webinars/model-predictive-control-overview-and-selected-applications (accessed Apr. 28, 2022).

[14] M. Cannon, "Efficient nonlinear model predictive control algorithms," *Annu. Rev. Control*, vol. 28, no. 2, pp. 229–237, Jan. 2004, doi: 10.1016/j.arcontrol.2004.05.001.

[15] R. Findeisen and F. Allgöwer, "An Introduction to Nonlinear Model Predictive Control," Jan. 2002.

[16] T. A. Johansen, "Introduction to Nonlinear Model Predictive Control and Moving Horizon Estimation".

[17] S. Yu, E. Sheng, Y. Zhang, Y. Li, H. Chen, and Y. Hao, "Efficient Nonlinear Model Predictive Control of Automated Vehicles," *Mathematics*, vol. 10, no. 21, Art. no. 21, Jan. 2022, doi: 10.3390/math10214163.

[18] "Nonlinear Model Predictive Control: Theory and Algorithms | SpringerLink." https://link.springer.com/book/10.1007/978-0-85729-501-9 (accessed Dec. 20, 2022).

[19] K. Numayr, "Comments on the understanding of the Virtual Work Method," *Int. J. Civ. Struct. Eng.*, vol. 2, Feb. 2012, doi: 10.6088/ijcser.00202030006.

[20] X. Xu and J. Ou, "Force identification of dynamic systems using virtual work principle," *J. Sound Vib.*, vol. 337, Nov. 2014, doi: 10.1016/j.jsv.2014.10.005.

[21] C.-C. Yang and J.-H. Wen, "A Hybrid Local Virtual Force Algorithm for Sensing Deployment in Wireless Sensor Network," in *2013 Seventh International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing*, Jul. 2013, pp. 617–621. doi: 10.1109/IMIS.2013.109.