Dynamic simulation model for different type of wheeled mobile robotic drives on leveled surface

P S Patel and R Trivedi

Department of Mechanical Engineering, Institute of Technology, Nirma University, Ahmedabad, 382481, India.

prashamspatel@gmail.com

Abstract. Dimensions of wheel base, payload weight and geometry, and type of drive configuration are some of the major factors to be consider during prototyping of a mobile robotic drive. For analysing the effect of such factors, a dynamic simulation model valid for any drive configuration using simple, omni or mecanum wheels is discussed in this paper. Sub-systems to take into account the inertia, motor characteristics normal force, friction force and wheel slip are discussed. Preliminary simulation results for path anticipation, for a control algorithm, for a four wheels omni drive robot with uneven mass distribution and variation of normal force due to acceleration of the robot are also shown in this paper. The simulation model is developed in Simulink/MATLAB.

Keywords: Dynamic simulation, contact forces, wheeled mobile robots

1. Introduction

Advance dynamic simulation models can be developed for simulating different mobile robotic drives by using various physical and mathematical relationship. Such a model could be useful for designing and prototyping of mobile robotic drives for specific applications. They can be used to analyze performance of robots in different ground conditions and payloads, with varying weight and size, with varying level of accuracy. Such a dynamic simulation algorithm with motion planning was developed by Seegmiller and Kelly in [3], however it did not take mathematical model of drive motor into consideration. Another model for 4-wheel-omni drive and track drive robots was discussed in [1] and [2] respectively. All of the above-mentioned works are for a specific type of drive system, however in practice a variety of mobile robot drive systems are available, for instance, Holonomic drives (Omni and Mecanum drives), Synchronous drive, and differential drive, Ackermann drive. Each having their own specific advantages in maneuvering, degrees of freedom, responsiveness, complexity of mechanism and so on and so forth. Thus, a single model can be very time effective when comparing between different kind of drive configurations.

In this paper a comprehensive method for dynamic simulation of all types of drive configurations possible with use of simple, omni and mecanum wheels on a leveled plane field is discussed. A simulink subsystem is developed which can be attached directly to different 3-d models in simulink,
with minor tweaks, to perform the simulation. The following part explains the contact force model which is used to simulate the effect of different forces acting on the body. The third part of the paper discusses DC-motor model and its combination with contact force model. Then the simulink model as a whole is discussed, which is followed by results of preliminary simulation. And finally, conclusion of this paper is presented.

2. Contact force model

2.1. Convention and nomenclature

The convention followed by the paper for main robot body frame and contact frame for a 4-wheel-omni drive robot, which will be further used, is shown in Figure 1. Here, counter clockwise rotation is considered as positive direction of rotation and a right-hand co-ordinate system is used.

A contact force frame is placed at the point of contact of wheel and ground. The orientation of the frame for simple and Omni-wheels is such that Y-axis of the frame is concentric to the axis of rotation of wheel and Z-axis is parallel to the Z-axis of the world frame [5]. And for mecanum wheel the Y-axis of the frame is at an angle of 45 degree with axis of rotation of wheel and perpendicular to the axis of rotation of beads of the wheel, and Z-axis of the frame is parallel to the Z-axis of world frame [4]. Here also, a right-handed co-ordination system for both wheels is used. If the wheel is broad, multiple frames can be placed along the line of contact of wheel and ground. In Figure 2 top view of contact frame of simple and mecanum wheel is shown and Table 1 shows nomenclature used in contact force model.

![Figure 1](image1.png)

**Figure 1.** Convention for 4-wheel-omni drive

![Figure 2](image2.png)

**Figure 2.** Contact frame: simple and omni-wheel (a) and Mecanum wheel (b)
Table 1. Nomenclature for contact force model

| Symbol | Variable                        |
|--------|--------------------------------|
| $F_n$  | Normal force                   |
| $K$    | Stiffness of contact surface   |
| $\delta$ | Deflection of contact surface |
| $D$    | Damping factor of contact surface |
| $\dot{\delta}$ | Relative speed of contact surface |
| $F_f$  | Friction force                 |
| $F_r$  | Rolling friction force         |
| $F_c$  | Coulomb friction force         |
| $F_s$  | Static friction force          |
| $\mu$  | Coulomb friction factor        |
| $C_r$  | Rolling friction factor        |
| $\dot{X}$ | Relative speed of body to ground |
| $Z$    | Deflection of bristles         |
| $\dot{Z}$ | Relative speed of bristles     |
| $\sigma_0$ | Stiffness of bristles        |
| $\sigma_1$ | Damping factor of bristles    |
| $\sigma_2$ | Viscous damping factor        |
| $V$    | Ground velocity of contact force along its X-axis |
| $V_r$  | Wheel slip velocity           |
| $\omega$ | Angular velocity of wheel     |
| $r$    | Radius of wheel               |

2.2. Normal force model

Normal force is calculated by using a penetration penalty model based on Linear Kelvin-Voigt Contact Model [9]. Mathematically the model can be represented as:

$$F_n = K \cdot \delta + D \cdot \dot{\delta}$$

(1)

Figure 3. Linear Kelvin-Voigt Contact Model
In simulink model shown in Figure 3, a dead zone block and a sign block is used to activate the model only when the frame is in contact with the ground, therefor Z-co-ordinate of contact frame with respect to world frame is less than or equal to zero. Two inputs to the model are Z-co-ordinate and velocity of frame along Z-axis of the world frame.

2.3. Friction force model

The application of tyre friction model in this paper is inspired from [6] and [7]. But unlike the LuGre friction model used in the previous mentioned works, the friction model used here is somewhere between LuGre and Dahl friction. Here, Stribeck effect is neglected by taking Stribeck velocity as zero (Vs=0) as in Dahl model, but value of parameters $\sigma_1$ (stiffness factor) and $\sigma_2$ (Micro-damping factor) are non-zero as in LuGre model. Figure 4 shows Simulink model for the Lugre friction and mathematically it can be represented as:

\[
F_c = \mu \cdot F_n \tag{2}
\]

\[
G\left(\dot{X}\right) = F_c + \left(F_s - F_c\right) \cdot e^{\left|\frac{\dot{X}}{V_s}\right|} \tag{3}
\]

\[
\dot{Z} = \dot{X} - \frac{\sigma_0 \cdot Z \cdot \dot{X}}{G\left(\dot{X}\right)} \tag{4}
\]

\[
F_f = \sigma_0 \cdot Z + \sigma_1 \cdot \dot{Z} + \sigma_2 \cdot \dot{X} \tag{5}
\]

For Vs=0,

\[
G\left(\dot{X}\right) = F_c \tag{6}
\]

\[
\dot{Z} = \dot{X} \left(1 - \frac{\sigma_0 \cdot Z}{F_c}\right) \tag{7}
\]

\[
F_f = \sigma_0 \cdot Z + \dot{X} \left[\sigma_1 + \sigma_2 - \frac{\sigma_1 \cdot \sigma_0 \cdot Z}{F_c}\right] \tag{8}
\]

As discussed by authors in [8] value of Z must always be less than or equal to $G(\dot{x})/\sigma_0$. Therefore, $\sigma_0 \cdot Z$ must always be less than $F_c$. In the model a dynamic saturation block is used whose limits are value of coulomb frictional force to replicate such effect. Here, a constant of the order $10^{-10}$ is required to be add to the normal force when calculating damping force. This is done so that when normal force is zero, the model does not produce error due to division by zero. $\dot{x}$ is equal to $V_r$ in this model. The sign of this term is opposite to the sign of the term $V_r$, where $V_r$ is described as:
\[ Vr = r \cdot \omega - V \]  \hfill (9)

Inputs for this model are normal force and relative velocity (Vr). Rolling resistance of the wheel is also taken account of, and its direction is same as that of Vr. This force should be applied along the X-axis of the contact frame for all three kinds of wheels. For calculation of friction in Y-axis there are different models for different kind of wheel:

1. Simple wheels:

   model will be same as above, only ‘\( \dot{x} \)’ will be equal to the negative of the relative velocity (V) between the contact frame and ground along the Y-axis of the contact frame.

2. Omni and Mecanum wheels:

   Only rolling friction will act along the Y-axis of the frame in the opposite direction of motion. Rolling friction force is calculated according to the formula give below:

\[ F_f = C_R \cdot F_n \]  \hfill (10)

**Figure 4.** LuGre friction model

**Figure 5.** Complete contact force model

2.4. Complete contact force model

Figure 5 shows the complete contact force model, as can be seen two transform sensor blocks are used to acquire values for current velocity and position of the contact frame. Both blocks have contact frame as follower frame and world frame as base frame. The first sensor block measure relative position and speed of contact frame along Z-axis of world frame, therefor base frame as the measurement frame. And second block is used to measure relative speed of contact frame and world frame along X and Y axis of the contact frame, therefor follower frame as measurement frame.

All the forces calculated are given to external force and torque block with contact frame as reference. For simple and omni wheels the force along X-axis is multiplied by wheel radius to be converted into load torque and is passed on to DC-motor model. This is valid as X and Y-axis of contact frame and wheel are coincident. However, for mecanum wheel the component of force along X and Y axis along the X axis of the wheel (which is at 45° counterclockwise from contact frame as show in Figure 2) are added and multiplied by wheel radius to calculate load torque. A unit delay is applied before passing the load torque as to avoid the formation of algebraic loop.
3. DC-motor model

The Model was created by augmentation of the inbuilt DC- motor model. An ideal torque source block was added to account for the load torque on the motor. An ideal voltage source block was also added to reverse the polarity of applied voltage on the motor, by signaling to REV port of the H-bridge block above its threshold voltage.

The model has three physical signals as inputs, a PWM duty cycle signal and other two signals, for controlling output voltage of ideal voltage source block and load torque on motor applied by ideal torque source block. A single output physical signal, of angular velocity of motor, is given by the model. Figure 6 shows the DC-motor model in Simulink.

![Figure 6. DC-motor model](image)

![Figure 7. combined contact force and motor model](image)

As shown in Figure 7, both the contact force and DC-motor model are combined forming a loop, PWM signal is the only simulink input and speed of the motor is integrated and converted into angular position and exported to the revolute joint block for accounting the moment of inertia of the wheel. An absolute value of PWM signal is given to DC-motor as PWM block only takes positive input. And negative of the sign of PWM signal is passed as input to ideal voltage source block to vary the polarity of the motor voltage. Other two physical connection are of base and follower frame for contact force model.

4. Complete model of the robot

The geometry of the robot can be generated in any CAD software and further exported to MATLAB by available plug-ins, here the SolidWorks was used to generate model. Appropriate degrees of freedom were given to the CAD model for Simscape to automatically generate required joint types and mechanism, for instance, revolute joint for wheel and steering mechanism for Ackerman and synchronous drive should be properly constructed in CAD software for Simulink to automatically generate required joint and mechanism.

For control of the robot as seen in Figure 8, a MATLAB function block has been used to write control algorithm for robot to follow desired trajectory. Here, a control algorithm for 4-wheel-omni robot was written to move along a sine-wave function. The required wheel velocity is to be represented from -1 to 1, -1 and 1 being maximum wheel angular velocity in negative and positive direction respectively and 0 being no angular motion of wheel. Here, the angular velocity output of the transform sensor block along Z-axis of the world has been integrated to get the current angular
position of the robot. And before passing the measurements of transform sensor to MATLAB function block a unit delay is given to avoid the formation of algebraic loops. The inputs for the MATLAB function block, such as co-ordinates and angular position of the robot are derived from a transform sensor block with world frame as base and main body of the robot as follower. 6-degrees of freedom were given to the robot body with respect to world by introducing 6-DOF joint block between world frame and robot body.

Two robot models were developed, a 4-wheel-omni robot which had an unsymmetrical weight distribution (show in Figure 9), thus having different load on different wheels and a 4-wheel-omni robot with even weight distribution. A suspension system with just one revolute joint between two pair of wheels was added to make sure that every wheel remains in contact with ground after micro-deflections simulated by the normal force model. Also, gravitational force was simulated by the mechanical configure block. The value of all the constant parameters used are in MKS system and are mention in the Table 2.

![Figure 8. complete 4-wheel-omni model](image)

![Figure 9. 3-d model of the robot with unsymmetrical weight distribution](image)

| Variable | Value          |
|----------|----------------|
| $K$      | $-1.5 \times 10^6$ |
| $D$      | 1500           |
| $\mu$   | 1.1            |
| $C_r$    | 0.005          |
| $\sigma_0$ | 100000        |
| $\sigma_1$ | 0.001         |
| $\sigma_2$ | 0.001         |
| $r$      | 0.075          |
5. Preliminary results

5.1. Steady state normal force and depression

This simulation was performed on a model of 4-wheel-omni drive robot having uniform weight distribution and zero net force along X and Y direction. The graphs showing normal force on each wheel and depression of frame (ideally to be 0.05m above ground) of the main-body of the robot with respect to time are shown in Figure 10 and 11. Each wheel had identical graph for normal force, due to uniform weight distribution.

As can be seen from the Figure 10 and Figure 11 both normal force and depression become almost constant after 0.1sec of simulation with final value of normal force oscillating between 72.5N and 70.9N, and a depression of 0.049952m. Thus, when simulating robot motion, it is better to start the motion after 0.1sec of starting of the simulation.

5.2. straight line motion

The model with symmetrical weight distribution was simulated to move along X-axis of the world frame and normal force acting on wheel-1 and wheel-2, according to the Figure 1, was plotted with respect to time.

As can be seen from the graph in Figure 12 and Figure 13 , the normal force varies with time and finally becomes steady. This change in normal force from an average value of 55N to 72N and 89N to 72N, in wheel 1 and wheel 2 respectively, is due to the moment produced by the friction force acting on the contact frame. Initially, when there is difference between the linear velocity of wheel and that of robot, the moment developed due to friction force lifts wheel-1 upwards (decreasing the normal force) and pushes wheel-2 downwards (increasing the normal force). And as gradually the velocity difference decreases, so does the friction force and finally the effect of the moment due to friction force is negligible and normal forces acting on the wheels are again equal.
5.3. Sin wave path following

For this simulation, model of 4-wheel-omni robot with uneven weight distribution was used. A sine wave path was to be followed by the robot. Figure 14 shows the code used for control, this code is called at every time step by the Simulink solver:

```
function [v1,v2,v3,v4] = fcn(za,zw,y,x,t)
    if t<0.1
        v=0;
    else
        v=38; %0<v<64
    end
    ha=-90+atan2d((sin(x+0.001)-y),0.001)+rad2deg(zw);
    Hda=0;
    kp=960;
    kd=512;
    v1=(fix(-v*0.707*sin(ha)-v*0.707*cosd(ha)-(deg2rad(Hda)-za)*kp-zw*kd))/64;
    v2=(fix(v*0.707*sin(ha)+v*0.707*cosd(ha)-(deg2rad(Hda)-za)*kp-zw*kd))/64;
    v3=(fix(v*0.707*sin(ha)+v*0.707*cosd(ha)-(deg2rad(Hda)-za)*kp-zw*kd))/64;
    v4=(fix(v*0.707*sin(ha)-v*0.707*cosd(ha)-(deg2rad(Hda)-za)*kp-zw*kd))/64;
end
```

**Figure 14.** Main drive code

‘ha’ is the heading angle which gives the desired direction of motion of the robot, ‘hda’ is the required angle of orientation of the robot with respect to ground about its z-axis. A simple PD-control is used for maintaining ‘Hda’, by using ‘za’ and ‘zw’ which are current angular orientation and rate of change of orientation of the robot with respect to ground about its z-axis respectively. A condition to start the motion after 0.1sec of starting of the simulation has also been implemented.

The direction of motion is calculated by finding slope of the line connecting current position of the robot and point on the sine wave which has abscissa 0.001 greater than abscissa of current position of the robot. Then adding ‘za’ for countering the effect of change in orientation of the robot. And finally subtracting 90° degrees to get angle according to the conventions discussed earlier.

\[
hda = \tan^{-1}\left( \frac{\sin(x+0.001) - y}{(x+0.001) - x} \right) + za - 90^\circ \quad (11)
\]

The sensors were considered ideal and a motor driver which has a total of only 64 different PWM duty cycles (starting from 0% to 100% in an increment of 1.5625%) was simulated. The Figure 15 and Figure 16 shows the actual path followed by the robot versus the desired path and angular position of the robot with respect to ground about its Z-axis in radians, respectively.

**Figure 15.** Actual path verses given path

**Figure 16.** Angular position of robot about its Z-axis
6. Conclusion

The preliminary results show that task of trajectory anticipation can be easily performed by the model. The effect of mass distribution, which is reflected on normal force acting on the wheel and which eventually effects the motion of the robot can also be analyzed. Apart from this new control algorithm specific to the drive can also be tested using this model. The amplitude of oscillation of the normal force seen in the graph 9, 11 and 12 can be reduced by lowering the value of maximum time step and relative tolerance of the solver. However, it will increase the computational time taken by the solver. Some miscellaneous model configuration parameters to be taken care of are given below:

- The solver selection should be set on variable step-size, auto-select.
- Maximum step size should not be more than 0.001.
- Zero crossing should be enabled for all, with adaptive setting on.

The model can be further improved to also simulate uneven surfaces by adding multiple contact frame along the circumference of the wheel with their X-axis tangential to circumference and Z-axis pointing towards the center. And the normal force is to be applied accordingly to the penetration in the surface whose Z-coordinate is a function of X and Y co-ordinates. The electrical model can also be augmented by adding models for different kinds of motors such as BLDC and induction motors. And even noise can be added to replicate effect of imperfect sensors.

References

[1] E. Hashemi, M. G. Jadidi and O. B. Babarsad, "Trajectory planning optimization with dynamic modeling of four wheeled omni-directional mobile robots," 2009 IEEE International Symposium on Computational Intelligence in Robotics and Automation - (CIRA), Daejeon, 2009, pp. 272-277, doi: 10.1109/CIRA.2009.5423195.
[2] Amezquita-Semprun, Kendrick & Del Rosario, Manuel, Jr & Chen, Peter. (2018). Dynamics Model of a Differential Drive Mobile Robot Towing an Off-axle Trailer. International Journal of Mechanical Engineering and Robotics Research. 7. 583-589. 10.18178/ijmerrr.7.6.583-589.
[3] Seegmiller Neal & Kelly Alonzo. (2015). Modular Dynamic Simulation of Wheeled Mobile Robots. Springer Tracts in Advanced Robotics. 105. 75-89. 10.1007/978-3-319-07488-7_6.
[4] N. Tlale and M. de Villiers, "Kinematics and Dynamics Modelling of a Mecanum Wheeled Mobile Platform," 2008 15th International Conference on Mechatronics and Machine Vision in Practice, Auckland, 2008, pp. 657-662, doi: 10.1109/MMVIP.2008.4749608.
[5] Balakrishna, R. & Ghosal, Ashitava. (1995). Modeling of Slip For Wheeled Mobile Robots. IEEE Transactions on Robotics and Automation. 11. 126 - 132. 10.1109/70.345944.
[6] Liang, Wei & Medanic, Jure & Ruhl, Roland. (2008). Analytical dynamic tire model. Vehicle System Dynamics - VEH SYST DYN. 46. 197-227. 10.1080/00423110701267466.
[7] Fan, Xiao-bin & Fan, Bing-xu & Wang, Feng & Xia, Qin-sheng & Wang, Ping-an. (2016). Tire/wheel torsional dynamic behaviour and road friction coefficient estimation. Journal of Vibration Engineering. 18. 10.21595/jve.2016.16711.
[8] C. Canudas de Wit, H. Olsson, K. J. Astrom and P. Lischinsky, "A new model for control of systems with friction," in IEEE Transactions on Automatic Control, vol. 40, no. 3, pp. 419-425, March 1995, doi: 10.1109/9.376053.
[9] P. Flores and H.M. Lankarani, Contact Force Models for Multibody Dynamics, Solid Mechanics and Its Applications 226, DOI 10.1007/978-3-319-30897-5_3