Modelling of Mobile Robot Dynamics

Edouard Ivanjko
University of Zagreb

Ivan Petrovic
University of Zagreb

MODELLING OF MOBILE ROBOT DYNAMICS

Edouard Ivanjko[1], **Toni Petrinić**[2], **Ivan Petrović**[1]

[1]University of Zagreb, Faculty of Electrical Engineering and Computing
10000 Zagreb, Unska 3, Croatia
[2]HEP d.d., 10000 Zagreb, Ulica grada Vukovara 37, Croatia

edouard.ivanjko@fer.hr(Edouard Ivanjko)

Abstract

This paper presents two approaches to modelling of mobile robot dynamics. First approach is based on physical modelling and second approach is based on experimental identification of mobile robot dynamics features. Model of mobile robot dynamics can then be used to improve the navigational system, especially path planing and localization modules. Localization module estimates mobile robot pose using its kinematic odometry model for pose prediction and additional sensor measurements for pose correction. Kinematic odometry models are simple, valid if mobile robot is travelling with low velocity, low acceleration and light load. Disadvantage is that they don't take any dynamic constraints into account. This leads to errors in pose prediction, especially when significant control signal (translational and rotational velocity reference) changes occur. Problem lies in the fact that mobile robot can't immediately change its current velocity to the desired value and mostly there exists a communication delay between the navigation computer and mobile robot micro-controller. Errors in predicted pose cause additional computations in path planning and localization modules. In order to reduce such pose prediction errors and considering that mobile robots are designed to travel at higher velocities and perform heavy duty work, mobile robot drive dynamics can be modelled and included as part of the navigational system. Proposed two modelling approaches are described and first results using a Pioneer 3DX mobile robot are presented. They are also compared regarding to complexity, accuracy and suitability of implementation as part of the mobile robot navigational system.

Keywords: Modelling, mobile robot, estimation, dynamic model.

Presenting Author's Biography

Edouard Ivanjko received his PhD in 2009. from Faculty of Electrical Engineering and Computing, University of Zagreb, Croatia where he is currently working. His main research interests are: mobile robotics, localization of mobile robots and environment modelling for mobile robots. He published 3 papers in international journals and 19 papers in proceedings of international conferences. He is a member of IEEE, KoREMA and Croatian Society for Robotics. He speaks fluent German and English.

1 Introduction

In recent years huge interest in autonomous industrial vehicles can be noticed. Control systems for such vehicles should take into account all disturbances which can appear during their missions. Their control systems should react quickly and adapt to changing environment conditions. Many "classical" methods for designing control systems (optimum-control methods, algebraic methods) for such vehicles require a physical system description together with all of its parameters. For accurate mobile robot system description it's necessary to generate an appropriate dynamic model. The dynamic model allows considering such properties as: mass, inertia, friction forces, centrifugal force, torque, etc. Such models are built in order to better understand the structure and operation of the controlled mechatronic system. Creation of such a model becomes even more important if high complex systems have to be developed. Preparation and application of needed dynamical model allows early detection of flaws and mistakes in the description (model) of the real system. Their modification is simpler and less expensive in the virtual construction stage in comparison with physical prototypes.

Knowledge about mobile robot dynamics is very important for planning feasible mobile robot trajectories. First a path consisting of a series of poses is generated in the path planning module. Such a path is generated regarding to mobile robot dimensions in the sense that the mobile robot can traverse free workspace underlying planned path. In this case dynamic properties are not important. Desired path following is only done in geometric space and path planning criterion is that the mobile robot doesn't collide with any stationary obstacles. Typical path planning algorithms for such applications are the A* and D* algorithms [1].

Things change when moving obstacles are part of mobile robot workspace. They can be people or other mobile robots or other moving machines. In these situations a certain pose must be reached by the mobile robot in a certain time frame to avoid collision. In such cases mobile robot motion velocity has to be constantly altered in an appropriate way that mobile robot can avoid collision with a moving obstacle. Only in that way a collision free motion in a workspace with moving obstacles can be guaranteed [2]. During trajectory planning phase mobile robot dynamic properties are used to create a set of velocity profiles that can be performed by the mobile robot in a safe manner. In this process mobile robot velocity limitations must be respected with preserved trajectory curvature [3].

Mobile robot dynamical model is very important in cases when mobile robot velocities generate forces which influence can't be neglected during movement. A good example is mobile robot soccer where used mobile robots are small and velocities are significant compared to their size and mass. Typical shape is a cube of about 7.5 (cm) size and velocities can be faster than 2 (m/s). Small size and mass combined with such velocity values can cause slippage of drive wheels or turn overs in a curvature trajectory. Slippage can especially occur when significant velocity changes (big acceleration or deceleration values) are requested from the mobile robot. Appropriate control strategies and trajectories need to include mobile robot dynamics properties [4].

Recent published research also uses mobile robot dynamics to cope with workspace floor characteristics. In [5] authors developed a two-level mobile robot motion control strategy that can cope with different workspace floor characteristics. Low level part is a classical wheel velocity controller whereas high level part uses measured wheel velocities to adapt generated trajectory if there are significant differences from desired values. For this purpose authors modelled the mobile robot as a rigid body that rolls on two drive wheels and one castor including velocity changes constraints by the used actuators.

Mobile robot dynamic model can be also used to improve estimated pose in the localization module. This module mostly uses a kinematic model to predict mobile robot pose using appropriate control input values. In this case control input consists of translational and rotational velocities. Predicted pose is then corrected using additional sensor measurements. A typical example is usage of non-linear Kalman filters for this task [6]. Quality of so estimated pose depends on used kinematic model accuracy, used additional sensors and workspace model (map) accuracy. Kinematic model accuracy has a significant influence in such a framework.

Control input for kinematic model is usually computed in a navigational computer connected via a communication link with a low level micro-controller. The micro-controller handles drive wheels velocity control and their current velocity measurement. Velocity measurement is then returned to the navigational computer and used for pose estimation or path planning computations. Control input values of the navigational computer can differ from values that the micro-controller currently uses for drive wheels velocity control. Reasons are communication link delay, mobile robot dynamic properties and used drive mechanical characteristics. Mechanical characteristics include influence of friction, backlash, etc. Result is that used kinematic model predicts mobile robot movement, while micro-controller hasn't received movement command yet or predicts a mobile robot movement that can't be performed by the used mobile robot. So, there exists a constant prediction error that can't be taken into account by means of calibration or path planning restrictions.

It would be beneficial to model such features and include them in the pose prediction step as part of the localization module. Also path planning could be improved, especially the moving obstacle avoidance part when generated path is altered in time space i.e. a trajectory is created. To do this mobile robot physical properties have to be examined and their influence on interesting variables has to be determined. In case of mobile robot navigation, crucial variable is its pose which is predicted i.e. estimated using drive wheel velocities. Input variables are velocity references and appropriate model should use them too as an input to provide current mobile robot velocity value on its output. These velocity values can then be used by the localization and other modules. This would be the first step or model creation. Second step would include validation of obtained model. It isn't good to validate the model on real mobile robot in its working environment due to danger of damage. More preferable is simulation testing where velocity data from mobile robot experiments are used. Such a way is used in this article also.

This paper presents two approaches to modelling dynamic mobile robot features including influence of mechanical drive characteristics. First approach is based on making a physical model of mobile robot body and components used for velocity control like velocity controller, motor, gearbox, etc. Second approach is based on experimental fitting of recorded mobile robot velocity data regarding reference velocity data. Both models are validated using velocity data recorded using a Pioneer 3DX mobile robot.

The rest of the paper is organised as follows. Second section contains description of both modelling approaches. Third section gives an overview on models implementations in MATLAB/SIMULINK. After that obtained results are given followed with models comparison. Paper ends with conclusion and description of future work on this topic.

2 Modelling approaches

As mentioned above, this article presents two approaches to mobile robot dynamics modelling. First steps needed for physical modelling are described and secondly steps taken for experimental identifications are described. Both models need to take into account maximal values of rotational and translational velocities including maximal rotational and translational acceleration and deceleration values. These model features can be easily taken into account by using saturation and slope limitation functions.

2.1 Physical modelling

Used Pioneer 3DX robot is a two wheeled differential drive robot, where each wheel is driven independently. Forward motion is produced by both wheels driven at the same rate, turning right is achieved by driving the left wheel at a higher rate than the right wheel and vice versa for turning left. This type of mobile robot can turn on the spot by driving one wheel forward and second wheel in the opposite direction at same rate. Third wheel is a castor wheel needed for mobile robot stability. Drive wheels are equipped with encoders and their angular velocity readings become available through simple routine calls. Kinematic model of a differential drive mobile robot can be found in [7] and geometrical dependencies are given in Fig. 1, where r is drive wheel radius (mm), v_L and v_R are left and right drive wheel velocities, respectively (mm/s), x and y present mobile robot position in cartesian coordinates in (mm), and b is axle length between drive wheels (mm).

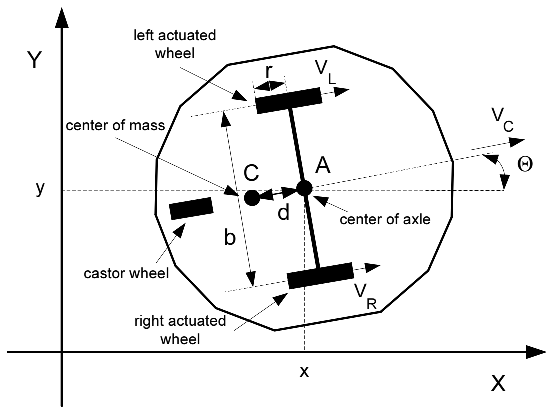

Fig. 1 Geometrical dependencies of a differential drive mobile robot.

Dynamic motion equation can be derived using Euler-Lagrange formulation [8, 9]:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_i}\right) - \frac{\partial L}{\partial q_i} = Q_i, \tag{1}$$

where L stands for difference of kinetic, T, and potential, U, energy, q_i stands for generalized coordinate, and Q_i stands for generalized force that acts on the mechanical system.

Under assumption that mobile robot moves only on a plane surface, potential energy of robot is zero ($U = 0$) and we have to find only kinetic energy of the mobile robot. Kinetic energy of the whole structure is given by the following equation:

$$T = T_t + T_r + T_{rwr}, \tag{2}$$

where T_t is kinetic energy of mobile robot translation, T_r is kinetic energy of mobile robot rotation, and T_{rwr} is kinetic energy of rotation of wheels and rotors of DC motors, all in $(kgmm^2/s^2)$ respectively. Values of introduced energy terms can be expressed by following equations:

$$T_t = \frac{1}{2}mv_c^2 = \frac{1}{2}m(x_c^2 + y_c^2), \tag{3}$$

$$T_r = \frac{1}{2}I_A\dot{\Theta}^2, \tag{4}$$

$$T_{rwr} = \frac{1}{2}I_0\dot{\Theta}_R^2 + \frac{1}{2}I_0\dot{\Theta}_L^2, \tag{5}$$

where m is the mass of entire mobile robot (kg), v_c is linear velocity of the mobile robot's center of mass C (mm/s), I_A is the moment of inertia of entire mobile robot with respect to the point A $(kgmm^2)$, Θ is mobile robot orientation (rad), $\dot{\Theta}$ is mobile robot rotational speed (rad/s), I_0 is the moment of inertia of combined drive motor (rotor) and wheel $(kgmm^2)$, and

$\dot{\Theta}_R$, and $\dot{\Theta}_L$ are angular velocities of the right and left drive wheel, respectively (rad/s).

Components of the velocity of point A, can be expressed in terms of $\dot{\Theta}_R$ and $\dot{\Theta}_L$:

$$\dot{x}_A = \frac{r}{2}(\dot{\Theta}_R + \dot{\Theta}_L)\cos(\Theta), \tag{6}$$

$$\dot{y}_A = \frac{r}{2}(\dot{\Theta}_R + \dot{\Theta}_L)\sin(\Theta), \tag{7}$$

$$\dot{\Theta} = \frac{r(\dot{\Theta}_R - \dot{\Theta}_L)}{b}, \tag{8}$$

where $\dot{x}_A$ presents velocity of point A in direction of the X-axis (mm/s), and $\dot{y}_A$ presents velocity of point A in direction of the Y-axis (mm/s).

Components of the velocity of point C considering velocity of point A are now:

$$\dot{x}_C = \dot{x}_A + d\dot{\Theta}\sin\Theta, \tag{9}$$

$$\dot{y}_C = \dot{y}_A - d\dot{\Theta}\cos\Theta, \tag{10}$$

where d is the distance between points A and C in (mm), and $\dot{x}_C$ presents velocity of point C in direction of the X-axis (mm/s), and $\dot{y}_C$ presents velocity of point C in direction of the Y-axis (mm/s).

Total kinetic energy of the mobile robot can be calculated in terms of $\dot{\Theta}_R$ and $\dot{\Theta}_L$:

$$\begin{aligned} T(\dot{\Theta}_R,\dot{\Theta}_L) \;=\; & \left(\tfrac{mr^2}{8}+\tfrac{(I_A+md^2)r^2}{2b^2}+\tfrac{I_0}{2}\right)\dot{\Theta}_R^2 \\ & +\left(\tfrac{mr^2}{8}+\tfrac{(I_A+md^2)r^2}{2b^2}+\tfrac{I_0}{2}\right)\dot{\Theta}_L^2 \\ & +\left(\tfrac{mr^2}{4}-\tfrac{(I_A+md^2)r^2}{b^2}\right)\dot{\Theta}_R\dot{\Theta}_L. \end{aligned} \tag{11}$$

Now, Lagrange equations:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial\dot{\Theta}_R}\right) - \frac{\partial L}{\partial\Theta_R} = M_R - K\dot{\Theta}_R, \tag{12}$$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial\dot{\Theta}_L}\right) - \frac{\partial L}{\partial\Theta_L} = M_L - K\dot{\Theta}_L, \tag{13}$$

are applied. M_R and M_L are right and left actuation torques, respectively in $(kgmm/s^2)$ and $K\dot{\Theta}_R$ and $K\dot{\Theta}_L$ are viscous friction values of right and left wheel-motor systems, respectively in $(kgmm/s^2)$. Finally, dynamic motion equations can be expressed as:

$$A\ddot{\Theta}_R + B\ddot{\Theta}_L = M_R - K\dot{\Theta}_R, \tag{14}$$

$$B\ddot{\Theta}_R + A\ddot{\Theta}_L = M_L - K\dot{\Theta}_L, \tag{15}$$

$$A = \left(\frac{mr^2}{4} + \frac{\left(I_A + md^2\right)r^2}{b^2} + I_0\right), \tag{16}$$

$$B = \left(\frac{mr^2}{4} - \frac{\left(I_A + md^2\right)r^2}{b^2}\right). \tag{17}$$

2.2 Experimental identification

Number of needed values is much smaller in this case and it consist of maximal rotational and translation velocity including their maximal acceleration and deceleration values.

Other features can be obtained by creating appropriate experiments. To obtain this features, first critical velocity change cases have to be defined. Such cases are partly covered with the above mentioned maximal values. Other cases are, when mobile robot changes its translational or rotational velocity direction, when it starts its movement (step wise velocity reference change), and when velocity is constant (steady movement) or constantly changing (mobile robot is accelerating or decelerating). Figs. 2 to 5 show mobile robot reference and measured velocity relationship for mentioned cases. Only translational velocities are presented whence rotational velocity responses show similar behaviors. Velocity responses where obtained using a Pioneer 3DX mobile robot controlled with an application using 100 (ms) sampling time. Only for stand still area observation smaller sampling time was used (20 (ms)).

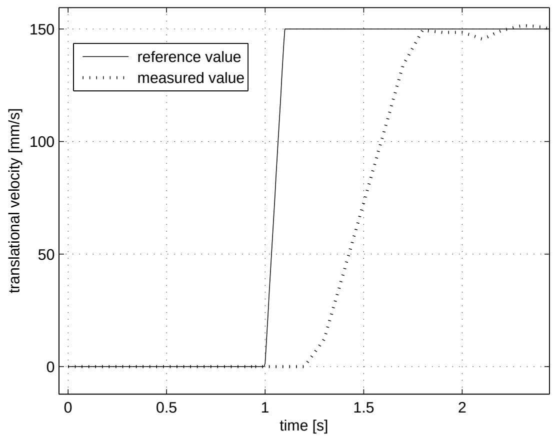

Fig. 2 Mobile robot velocity response in case of a step-like velocity reference change.

It can be seen that in case of a step-like velocity reference change (Fig. 2), mobile robot velocity response is similar to a ramp determined by maximal acceleration value. Such behavior is expected but there is also an additional time delay in the velocity response. It can be explained as a communication delay. Fig. 3 displays case of a constant reference. Mobile robot can hold desired velocity with influence of noise. On the right side

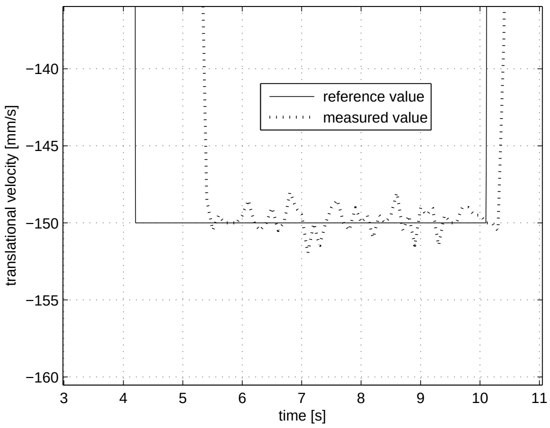

Fig. 3 Mobile robot velocity response in case of a constant velocity reference.

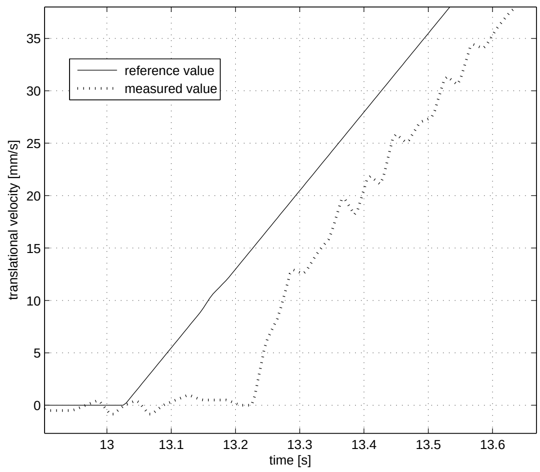

Fig. 4 Mobile robot velocity response in case of a ramp-like velocity reference change.

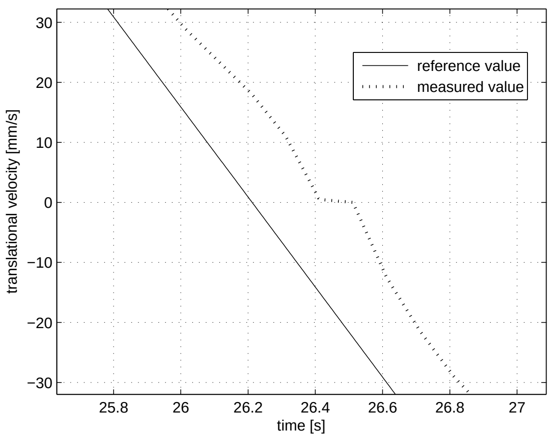

Fig. 5 Mobile robot velocity response in case of a ramp-like velocity reference change crossing stand still point.

of the figure, influence of the mentioned time delay can be observed. Fig. 4 displays case of a ramp-like velocity reference change. Also a time delay can be observed including a greater error when the ramp begins. One part of the greater error happens due to used drive controller influence and friction. Fig. 5 displays the case of a velocity direction change. An area where mobile robot stands still can be observed. Deceleration to stand still happens constantly like the reference but acceleration in the opposite direction continues after velocity reference reaches a certain value. This feature can be explained with influence of movement in previous direction and friction.

In order to reduce error between control input values used for velocity values prediction and true mobile robot movement values, appropriate dynamic model can be used. It has to take into account all mentioned cases. Additionally, model has to be simple so that it doesn't make any additional burden on the navigation computer. Also a small number of model parameters is preferable to enable a possibility of their on-line estimation. Communication delay can be modelled using transport delay, and behavior around standstill using a variable threshold. After reference value reaches threshold value, estimated velocity values start to rise.

Crucial model values that have to be obtained experimentally are communication delay and threshold at zero velocity value. Communication delay value is determined by comparing sent velocity reference value and measured velocity value in time space. Accordingly, each reference and velocity value had a corresponding time stamp. Obtained time delay value is 250 (ms) for translational part and 270 (ms) for rotational part. It has to be mentioned here that mobile robot controller works internally with a sampling time of 5 (ms), and sends out averaged velocity values every 100 (ms). This explains why obtained communication delay value isn't an integer multiple of the sampling time. For this reasons communication delay is modelled as a combination of integer delay and ZOH discretized first order transfer function. That means that delay of 270 (ms) in case of the translational part is modelled as delay of two discretization steps and the rest of 70 (ms) as a first order discrete transfer function with time constant of 100 (ms) and discretization time of 100 (ms).

Needed threshold value depends on the ramp slope, i.e. velocity acceleration or deceleration and its value is then computed from a dependency recorded from a series of experiments. Experiments where done for characteristic values of velocity change and a good enough approximation can be made using a straight line. Fig. 6 displays obtained threshold values for translational part of the model. Rotational part is obtained in an analog way.

3 Model implementation

This section describes how obtained models were implemented in MATLAB/SIMULINK. Both models are implemented with equal requirements. First requirement is of course best possible accuracy. Second requirement was that model can accept measurements obtained from a real mobile robot and compare velocity values for model accuracy validation. For the sake of

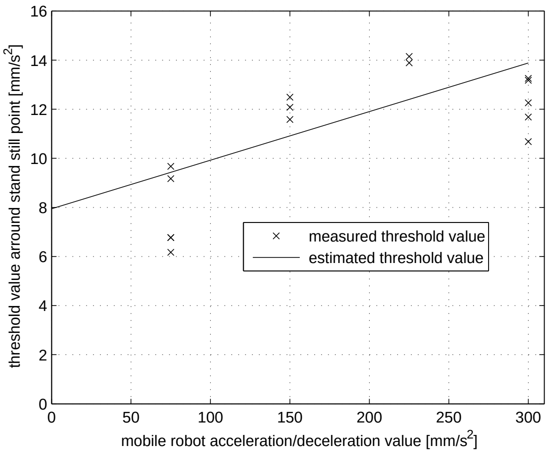

Fig. 6 Threshold value for translational part.

comparison, deviation of the estimated velocity from the measured one is computed including average and maximal error values.

3.1 Physical model

Pioneer 3DX drive system uses high-speed, high-torque, reversible-DC motors, each equipped with a high-resolution optical quadrature shaft encoder for precise position and speed sensing and advanced dead-reckoning. Motor parameters can be found in [10] and most important ones are given in Tab. 2. Mobile robot parameters, motor gear head ratios and encoder ticks per revolution can be found in [11] and most important ones are given in Tab. 1. It has to be noticed that values I_0 and I_A were estimated and the rest is taken from mobile robot manufacturer data.

Tab. 1 Mobile robot parameters

Parameter	Value	Unit
m	28.05	(kg)
r	95	(mm)
b	320	(mm)
d	57.8	(mm)
I_0	$9.24 \cdot 10^{-2}$	$(kgmm^2)$
I_A	$175 \cdot 10^3$	$(kgmm^2)$

Tab. 2 Mobile robot drive parameters

Parameter	Value	Unit
K_A	0.013	(A/V)
K_M	0.029	$(Nm/\sqrt{W})$
T_A	1.1	(ms)
T_M	8.5	(ms)
K	$35 \cdot 10^{-7}$	(Nms/rad)

Coordinates of mobile robot center of mass and moment of inertia with respect to the mobile robot center of mass were computed by separating used mobile robot on distinctive elements whose mass and pose inside mobile robot could be easily measured. It is assumed that mass of each element is concentrated in geometric center of that element. This is a good approximation because all elements have a symmetrical shape and constitution like batteries, wheels, motor with gearbox, case, etc. PI controller for drive wheel angular velocity is used. Integral time constant compensates dominant time constant of the velocity control loop. Proportional gain is chosen in order to damping factor of regulation loop be satisfied ($\xi = 0.9$). PI parameters were so $K_R = 11996$ and $T_I = 4.58$ (s).

Obtained model can be seen in Fig. 7. Model input variables are mobile robot rotational and translational velocity references. Their values and known mobile robot kinematic model with velocity and acceleration constrains are then used to compute left and right drive wheel speed references including time delay. These references are then used as input for left and right drive wheel speed controller. Coupling between left and right side is also modelled. End part of the model on the right side computes final mobile robot rotational and translational velocities. Number of encoder ticks per revolution equals 500.

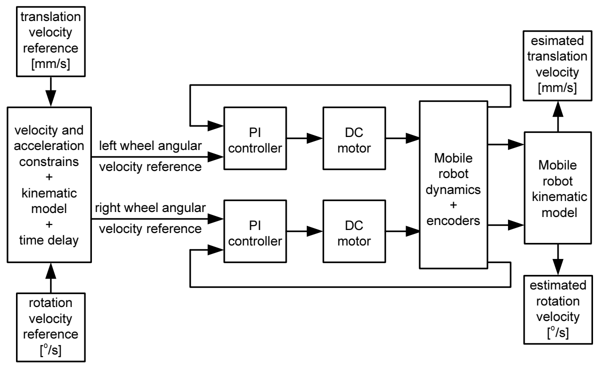

Fig. 7 Block scheme of proposed physical model.

Time delay is in this case difficult to model because amount of time needed for velocity data transmission and their evaluation in navigational computer or mobile robot micro-controller isn't documented. It can be heuristically determined and verified in simulation. So, time delay of 3 time steps (300 (ms)) was used.

3.2 Experimental model

Fig. 8 presents proposed experimental model block scheme consisting of parts described in subsection 2.2. It can be used for translational and rotational velocity part. Model input is generated velocity reference and output is estimated mobile robot velocity. To obtain estimated values of both, rotational and translational velocities, two models have to work in parallel.

According to model blocks in Fig. 8, proposed model can be implemented mostly using standard SIMULINK blocks. Only problematic part is the block labelled "Threshold at zero velocity". It has to influence the estimated velocity value only when mobile robot velocity is changing its direction. And then, only in the case when absolute velocity value is beginning to rise. This part is solved by detecting appropriate velocity change situation and then applying one of the following cases:

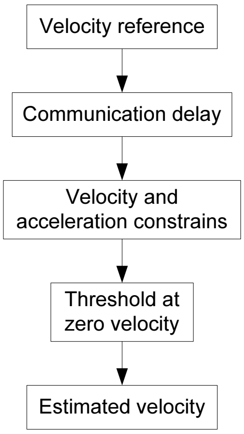

Fig. 8 Block scheme of proposed experimental model.

(i) estimated velocity value isn't changed, (ii) estimated velocity value is changed to zero. In that way separately deceleration to mobile robot stand still for the case of negative and positive velocity are detected. Estimated velocity changes are detected by comparing previous and current values. A wrap to zero block is used to make needed changes to estimated velocity value when estimated value is below the set threshold value.

4 Obtained results

In order to test proposed dynamic models two data sets obtained from a real Pioneer 3DX mobile robot were prepared. First data set consists of clearly separated critical velocity change cases. Second data set is taken from a navigation algorithm testing experiment. Navigation algorithm generated velocity references considering current obstacle situation in mobile robot environment and respecting mobile robot motion constraints. It presents a more realistic situation and is therefore better situated for described models accuracy comparison.

Tab. 3 Error values in case of rotational velocity estimation in $(^\circ/s)$ for first test values set

Error	Physical model	Experimental model
Maximal	4.31	4.59
Average	0.81	0.48

Tab. 4 Error values in case of translational velocity estimation in (mm/s) for first test values set

Error	Physical model	Experimental model
Maximal	18.8	12.4
Average	3.36	1.62

For the sake of a better model comparison using deviation values between this two data sets, average velocity values have to be known. First data set had maximal translational velocity of 300 (mm/s) and maximal rotational velocity of 50 $(^\circ/s)$. Second data set had maximal values of 600 (mm/s) and 63 $(^\circ/s)$, respectively.

Tab. 5 Error values in case of rotational velocity estimation in $(^\circ/s)$ for second test values set

Error	Physical model	Experimental model
Maximal	6.25	8.94
Average	1.19	0.92

Tab. 6 Error values in case of translational velocity estimation in (mm/s) for second test values set

Error	Physical model	Experimental model
Maximal	53.96	32
Average	6.77	4.66

Tab. 7 Error values in case of no model for second test values set in $(^\circ/s)$ and (mm/s)

Error	Rotation	Translation
Maximal	12.87	48.83
Average	2.13	14.76

Obtained values of maximal velocity error and its average value are given in Tabs. 3 to 6. For comparison Tab. 7 contains error values for the case with no model. Significant improvement can be observed. Apart error values, obtained velocity responses were also examined. They are given in Figs. 9 and 11. Only a part of recorded velocity values are presented for the sake of a better representation.

Figs. 10 and 12 present error between measured velocity and estimated velocity. Fig. 10 presents translational part for first data set and Fig. 12 rotational part for second data set. Other cases can be presented in a similar fashion.

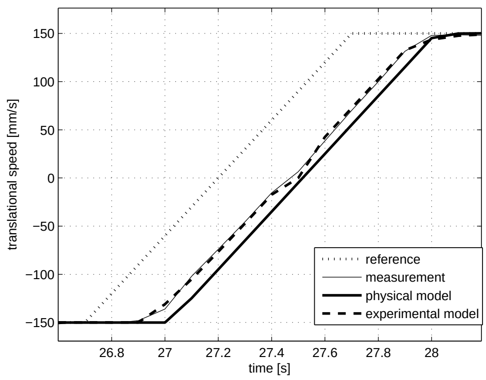

Fig. 9 Translational velocity response detail in case of the first data set.

5 Model comparison

As it was expected, second data set results with a less accurate estimation. One reason for the less accurate estimation is more frequent change of the velocity reference. Second reason are situations that can't be detected by an off-line model like drive wheel slippage or

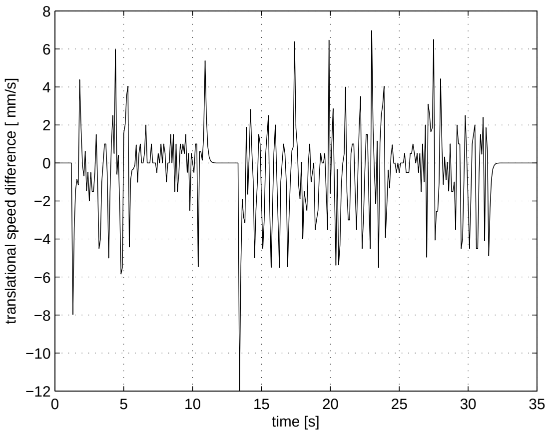

Fig. 10 Estimated translational velocity error for first test values set and physical model.

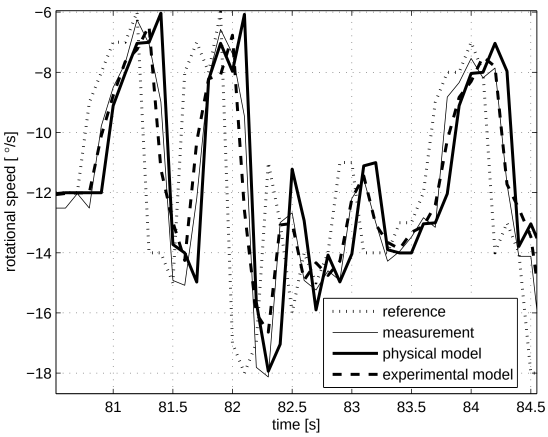

Fig. 11 Rotational velocity response detail in case of the second data set.

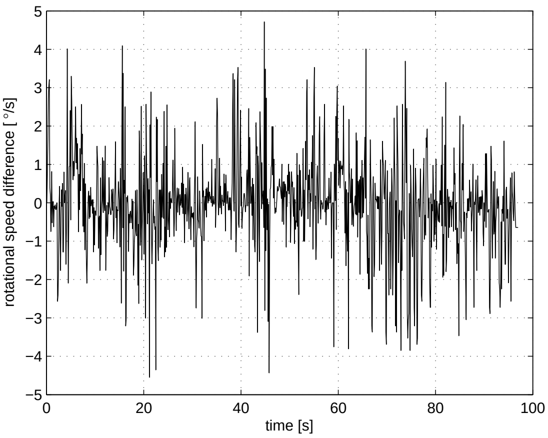

Fig. 12 Estimated rotational velocity error for second test values set and experimental model.

travelling over uneven floor. Mobile robot drive wheels are in such situations under random environment influence and their drive wheel rotation velocity doesn't reflect real mobile robot travelling velocity. But, their value can be used as a good base for real mobile robot travelling velocity estimation in combination with other sensors [12].

From data given in tables above, it can be observed that experimental model has smaller average error values which means it mostly better models examined mobile robot drive dynamics. Maximal error value is larger in the case of the experimental model. Such spikes are related to cases of significant velocity reference changes. In such cases internal states define partly velocity response and they are not included in the experimental model. That presents a drawback of this model. Physical model contains internal states of mobile robot drive system because it models the controller, DC motor, gearbox with encoders, and communication delay with appropriate transfer functions. So it performs better in cases where this internal states are significantly influencing velocity response. Currently this model doesn't include friction and backlash influence. Adding these features could improve this model accuracy but it would also complicate its implementation as part of mobile robot navigational system. So its more beneficial to use a simpler model with equal accuracy and, if possible, combine it with an on-line parameter estimation framework.

Another crucial aspect of the proposed dynamic models comparison is possibility of their implementation as part of mobile robot navigational system. Also possibility of making an on-line parameter estimation framework would be a good feature. In both cases only communication delay can be estimated on-line. One has to consider that only available measurement is mobile robot velocity. Regarding possibility of implementation experimental model is simpler and doesn't suffer from the accuracy loss when discretized.

6 Conclusion & future work

This paper presents two modelling approaches regarding mobile robot drive dynamics. First approach considers modelling every element of mobile robot drive system and corresponding control framework. Second approach models characteristic velocity change cases. It results in a simpler model that gives smaller average velocity error.

Model development consists of two phases. First phase consists of mobile robot examination to create an appropriate model concept i.e. modelling phase. This phase includes also creation of appropriate experiments so that all dynamic properties can be included in the model concept. Second phase includes firstly testing of the model concept i.e. simulation phase. Testing is done in MATLAB/SIMULINK simulation using experimental data from the first phase. Additionally data from a real mobile robot are used for simulation testing. Such an approach ensures an enough accurate model to be finally implemented as part of a mobile robot navigational system.

As mentioned both models are tested in MATLAB/SIMULINK environment using velocity data obtained from real mobile robot in real navigational conditions. First test results confirm an improvement in comparison to usage of velocity reference values for mobile robot motion prediction. When used by navigational system, mobile robot pose prediction could be more accurate and a more precise generated trajectory following can be assured.

So future work will go into direction of including this dynamic model into mobile robot navigational system (localization and path planing module) and expanding it with an on-line estimation framework.

7 Acknowledgement

This research has been supported by the Ministry of Science and Technology of the Republic of Croatia under grant No. 036-0363078-3018. Authors wish also to thank Marija Đakulović for help in providing needed velocity data from her navigation experiments for dynamic model testing.

8 References

[1] Anthony Stentz. Optimal and efficient path planning for unknown and dynamic environments. *International Journal of Robotics and Automation*, 10:89–100, 1993.

[2] M. Djakulović and I. Petrović. Dynamic window based approach to mobile robot motion control in the presence of moving obstacles. In *Proceedings of IEEE International Conference on Robotics and Automation - ICRA 2007, Roma, Italy, 10-14 April 2007*, pages 1986–1991, 2007.

[3] G. Oriolo, A. D. Luca, and M. Vendittelli. Wmr control via dynamic feedback linearization: Design, implementacion, and experimental validation. *IEEE Transactions on Control Systems Technology*, 10(6):835–852, 2002.

[4] Gregor Klančar, Borut Zupančič, and Rihard Karba. Modelling and simulation of a group of mobile robots. *Simulation Modelling Practice and Theory*, 15:647–658, 2007.

[5] A. Albagul and Wahyudi. Dynamic modelling and adaptive traction control for mobile robots. *International Journal of Advanced Robotic Systems*, 1(3):149–154, 2004.

[6] Edouard Ivanjko, Ivan Petrović, and Mario Vašak. Sonar-based pose tracking of indoor mobile robots. *AUTOMATIKA - Journal for control, measurement, electronics, computing and communications*, 45(3-4):145–154, 2004.

[7] Edouard Ivanjko, Ivan Komšić, and Ivan Petrović. Simple off-line odometry calibration of differential drive mobile robots. In *Proceedings of 16th International Workshop on Robotics in Alpe-Adria-Danube Region*, pages 164–169, Ljubljana, Slovenia, june 7-9, 2007.

[8] S. M. LaValle. *Planning Algorithms*. Cambrige University Press, New York, 2006.

[9] Jasmin Velagić, Bakir Lacević, and Nedim Osmić. *Motion Planning*, chapter Nonlinear Motion Control of Mobile Robot Dynamic Model, pages 529–550. InTech Education and Publishing, june 2008.

[10] Mobile robot drive motors parameters. www.allmotion.se/files/pdf/1707.pdf.

[11] Mobile robot construction parameters. http://www.activrobots.com/robots/p2dx.html.

[12] P. Goel, S. I. Roumeliotis, and G.S. Sukhatme. Robust localization using relative and absolute position estimates. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 1999.