



Journal Homepage: -www.journalijar.com

INTERNATIONAL JOURNAL OF ADVANCED RESEARCH (IJAR)

Article DOI:10.21474/IJAR01/13374
DOI URL: <http://dx.doi.org/10.21474/IJAR01/13374>



RESEARCH ARTICLE

COMPLETE LOCOMOTION ANALYSIS OF A SMALL DIFFERENTIALDRIVE MOBILE ROBOTIC PLATFORM

Murat Bakirci

Faculty of Aeronautics and Astronautics, Tarsus University, Mersin, 33400, Turkey.

Manuscript Info

Manuscript History

Received: 05 July 2021

Final Accepted: 09 August 2021

Published: September 2021

Key words:-

Differential Drive Mobile Robot
Velocity Propulsion Mode Locomotion
Test Stame-Machine Programming
Data-Driven System Identification State-
Space Model Linearization

Abstract

Mobile robots are becoming a part of more and more research areas due to their structural advantages and the increase in usage areas. Differential drive mobile robots are among the most preferred of this type of robots due to the convenience that they provide in engineering studies. It is quite important to test and structurally investigate primary parts such as motors and its sensors before being used in research applications. Before proceeding to further studies, it is very useful to do such tests as they may provide critical information about the robot which can be quite beneficial in terms of time, effort, and cost. To achieve this task, variety of methods are available in the literature such as structural locomotion tests and system identification. In the first part of this study, locomotion tests of a small mobile robot driven by servo motors and operating with a single microcontroller was performed using the velocity propulsion mode. Three different predefined routes were determined for the robot and the accuracy of the robot moving along these routes was investigated. Through these tests, it is aimed to examine how the robot interprets the basic movements such as rectilinear forward motion, curvilinear motion, and rotation around its own axis. The next part focuses on the system identification of the robot. A data-driven model for the robotic platform was developed to make a mobile robot perform the desired movements and system identification. Various step input commands were sent to the robot under consideration and the responses of the robot wheels to these inputs were examined. Circular movements were made to the robot with a range of velocity input values and the relationship between input and output was examined for both wheels of the robot. In the locomotion tests, it was observed that the robot completed the predetermined routes with minor errors. As a result of these tests, theoretical calculations and experimental results were compared and the reasons for the error parameters were discussed. Through system identification tests, it was observed that the right wheel of the robot was more consistent and produced closer to the expected value for each test performed.

Copy Right, IJAR, 2021,. All rights reserved.

Corresponding Author:- Murat Bakirci

Address:- Faculty of Aeronautics and Astronautics, Tarsus University, Mersin, 33400, Turkey.

Introduction:-

Mobile robots are extensively preferred robotic devices in many research areas due to their simple structure and wide variety of uses (Gonzales et al., 2018; Corrales et al., 2014; Bruder et al., 2019; Said et al., 2018; Chen et al., 2017). Parallel to the technological developments in hardware production, the mobilization of these robots has increased significantly. Thus, they are utilized effectively in many research fields from education (Howell et al., 2006; Mitnik et al., 2008) to space studies (Sasiadek, J., 2014). Another important advantage of these robots is that they can perform the jobs that humans cannot do (Nikolaidis et al., 2017; Decker et al., 2017; Carsten et al., 2007; Lingshuai et al., 2016; Rubio et al., 2019; Bevacqua et al., 2015). As the functions of the mobile robot used increase, its design and utilization becomes more difficult (Sheridan, T., 2016). For this purpose, it is a necessity to perform some robotic activities with the help of computers, such as tracking from a fixed distance (Yilmaz et al., 2006).

The difficulties encountered in mobile robots can be listed as multi-purpose design in general, needing a wide bandwidth for good communication, design constraints (Berceanu and Tarnita, 2012), incompatibilities in the integration of different software, excessive costs that increase depending on the detection methods used (Yilmaz et al., 2006). Due to the frequent and rapid changes in environmental characteristics, it is expected that mobile robots can reach the desired point and communicate with other robots without hitting the obstacles they encounter. Moreover, they must simultaneously control their own resources and be as self sufficient as possible, that is, they must be autonomous (Yannier and Sabanovic, 2012).

Differential drive mobile robots (DDMR) are the most preferred mobile robots. A generic DDMR has two wheels that can be driven independently via DC motors and a single frictionless tail wheel that balances the entire robot (Alves et al., 2018; Malu et al., 2014). Unlike stationary industrial robots, mobile robots require more complicated controls, higher performance and wide variety of sensors, depending on the functions they want to perform and the environment in which they will be operated, since they have free movement capability. DDMR type mobile robots are substantially designed to perform tasks that are difficult, dangerous and impossible for humans. Exploring space objects or deep oceans, cleaning narrow channels or for tasks that need to be repeated many times are just some of the usage areas of these robots. Since the total number of degrees of freedom is less than the number of inputs controlling the system, DDMRs are considered to be underactuated (Alves et al., 2018).

As with all robots, DDMRs should also be diagnosed before they can be utilized in specific applications. For this, it is necessary to make the system identification (SI) of the robotic system, that is, to recognize the structure of the robot (Kim and Tilbury, 2004; Schafroth et al., 2010; Soderstrom, 2010; Bruder et al., 2019; Chen et al., 2017). Undoubtedly, the behavior of robotic structures are quite complicated and therefore SI is often not an easy task to accomplish. Besides, robotic systems are generally nonlinear (Wang, 2013; Xu, 2011; Chung and Slotine, 2009) and the data received from these systems are mostly distorted by noise (Gupta et al., 2018; Ulmen and Cutkosky, 2010). Moreover, the availability of options for input/output observational data of the robotic system is often limited. The linear system approach is quite convenient to have a first impression of the robotic system (Ouyang et al, 2014; Liu and Chopra, 2010). Linearization of the system will facilitate the SI process as well as provide sufficient prior knowledge for the robotic system under consideration.

In this study, the velocity propulsion mode for a modified DDMR was examined. To test the robot, three different routes with precise known geometries were predefined. The robot was driven multiple times on each route until it performed the desired movement. After conducting driving tests on each route, the differences between theoretical calculations and experimental findings were determined. A system identification of the robot has also been performed. Identification tests were conducted for the robotic system using the linear model approach. Circular locomotion tests were performed for each wheel of the robot and the input/output relationship was investigated. Through data-driven model approach the behavior of the robotic system has been captured based on a set of observations via laboratory tests. For each particular tests, linear model parameters, which is gain and time constant, for the step input from 0 to the desired level and for the step input from the desired level to 0 were computed.

In the rest of the paper, the robot was introduced briefly and general information about the system was also given. After a brief explanation of the test methodology, the paper was concluded with the interpretation of the test results.

Methodology:-

Description of the Robotic Platform

The platform is structured as a differential drive robot with two wheels driven independently by servo motors, and a single frictionless tail wheel as shown in Fig. 1. The robot consists of a single body frame linked with vertical microcontroller panel that provide enough space on which to mount sensors and other equipments. The robot controller is equipped with electronics necessary to drive the motors, and provides expansion for additional capabilities. It has been enhanced with multiple sensors and a single embedded micro-processor that sample the sensors and can control the various motors. The torque of a servo motor is proportional to the current that flows through it, and the rotational speed of the motor is proportional to the voltage applied across its inputs. Two motor encoders that can be used to precisely measure the amount and rate of rotation of each motor shaft. This allows closed loop control of the distance traveled and speed.

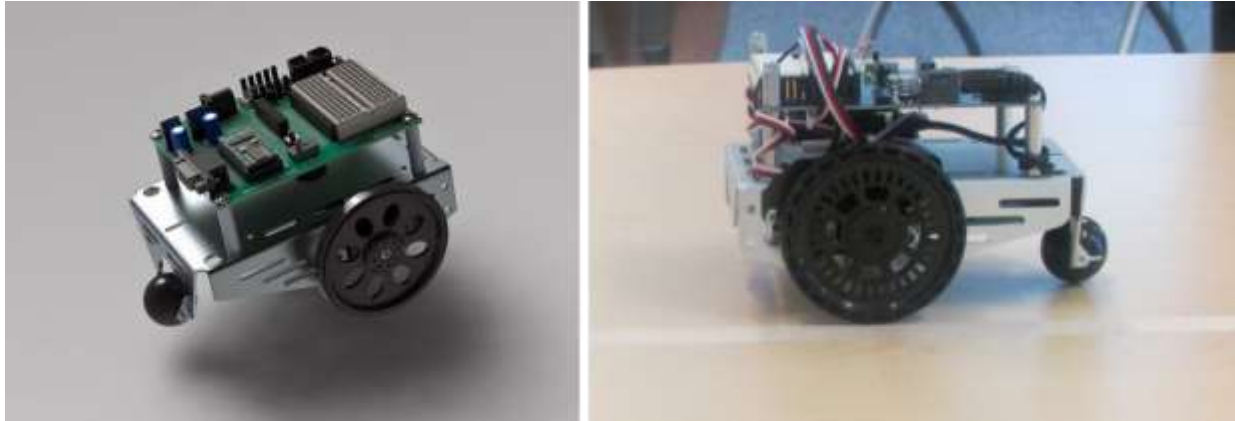


Figure 1:- CAD image of the plain robotic platform (left) and the modified robotic platform used in the laboratory experiments (right).

Locomotion Tests

To move the robot along pre-specified paths, specific programs need to be developed to issue necessary commands to the servo motors for appropriate amounts of time. There exist two different motor driving modes to execute particular robot movement tasks. The *effort mode* drives the motors at a percentage of their full capacity, while *velocity mode* automatically compensates for differences between the motors as well as varying levels of battery voltage. In this study, only the velocity mode of the robot was tested. In all portions of the tests, a series of tasks in sequence needs to be performed, i.e., move forward, turn, etc. A state machine algorithm structure has been preferred to achieve these tasks as shown in Algorithm 1 below.

Algorithm 1:- Structure of the state machine algorithm.

```

Int state; void
UserInit()
{
    (necessary initialization code) state = 1;
} void
UserLoop()
{ if (state == 1) {
    {1st step tasks}
    if (1st step is done){
        state = 2;
    }
}
else if (state == 2) {
    {2nd step tasks}
    if (2nd step is done) {
        state = 3;
    }
}

```

```

}           else {
            moveVelocity(0,0);
            }
}

```

The first test requires the robot travel along a straight line, reverse direction and then travel back along the same line reaching the point of origin. Fig. 2 depicts a top down view of the intended path. More specifically, the robot should start facing along the positive x direction, travel for 2 meters, and then pivot 180 degrees so it faces along the negative x direction. Then it travels for another 2 meters arriving at the same point it begun but facing the opposite direction. To ensure the robot has traveled exactly the desired distance the odometry is used to monitor travelled distance before stopping.

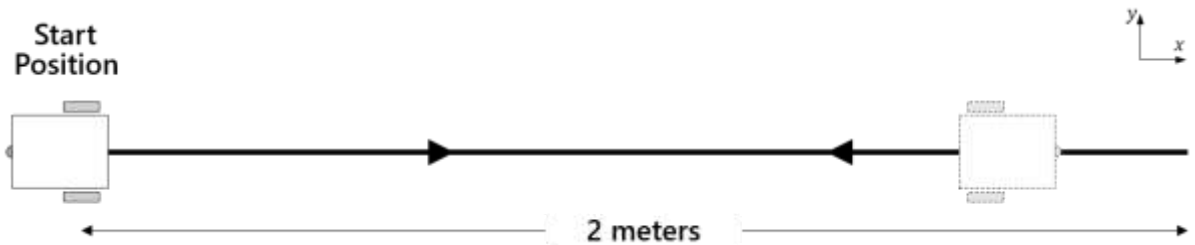


Figure 2:- Locomotion test path 1.

In the second test, a code which will make the robot travel along the shape shown in Fig. 3 is developed. The shape consists of a series of straight segments followed by a constant radius curve segment, which can be achieved by specifying different velocities for each wheel. Here, the primary goal is to have the robot arrive and stop as close to the start position as possible. It will appropriate to utilize odometry information to know when the robot moves across the trajectory transition points, i.e., transition from straight line to curved movement.

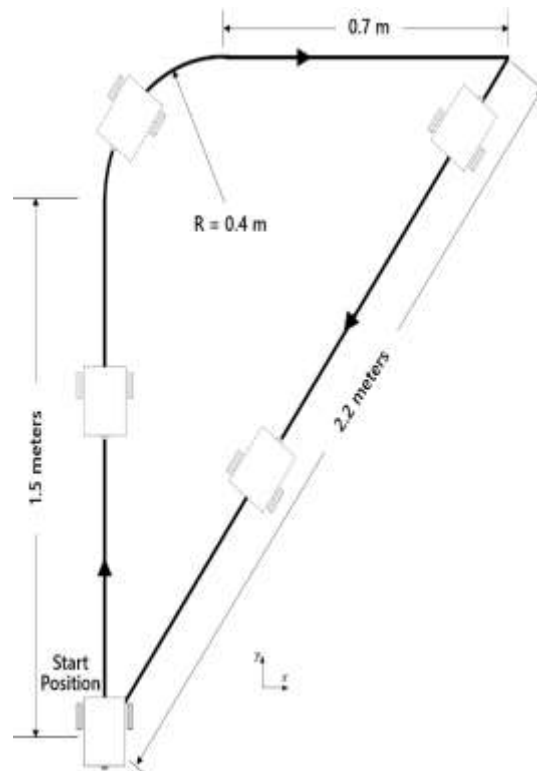


Figure 3:- Locomotion test path 2.

In the last test, the robot will be programmed to track the shape as shown in Fig. 4. The goal is for the robot to trace the pattern and stopping at the end location. A differential drive robot, such as the robot used in this study, will trace

a constant radius curve when the left and right wheel velocities are different. It will trace a straight line if the wheel speeds are the same. The radius of the circle traced, when the speeds are different, is given by the following equation:

$$R = \frac{w (V_L + V_R)}{2 (V_L - V_R)} \tag{1}$$

where R is the radius, W is the wheelbase of the robot, and V_L and V_R are the left and right wheel speeds respectively.

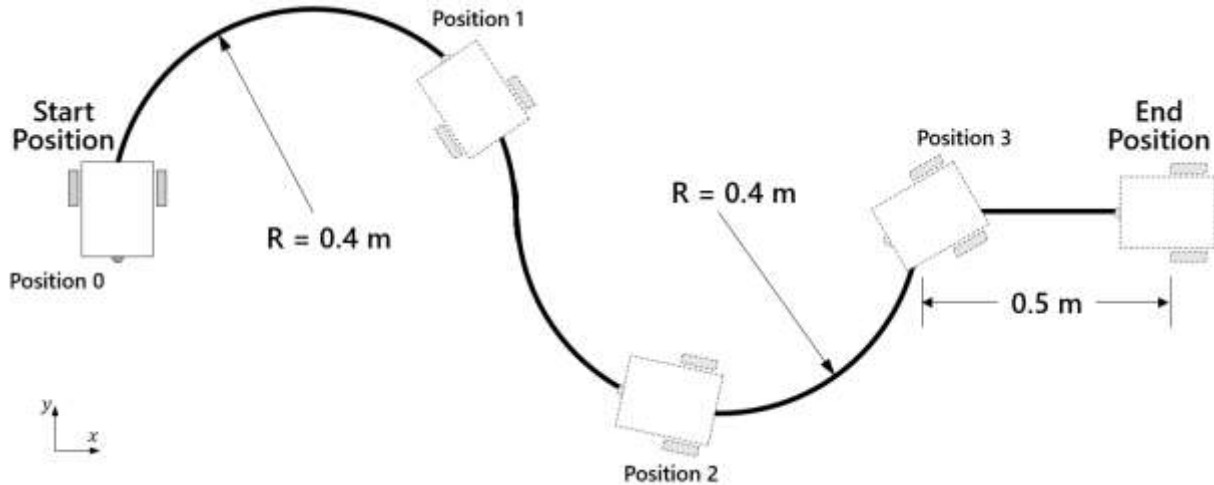


Figure 4:- Locomotion test path 3.

The distance traveled depends heavily on the type of floor materials. Solid floors, such as concrete or wood, will allow the robot to travel the longer distance because of lack of friction. Carpeted floors will cause friction and may prevent the robot from even moving at low effort values. Floors with tile or other grooves will further disrupt the movement.

System Identification

System identification can become a quite complicated process for most robotic platforms due to the conditions such as nonlinearity, extremely noisy data, etc. Nonlinearity, for instance, requires precise computation which may not even converge to produce desired results. In such cases it is quite convenient to use the linear model approach because such systems can simply be modeled by the identification of *gain* and *timeconstant*. The input for the mobile robotic system is the command sent to a wheel and the output of the system is the speed of that wheel. For a given gain, G , and time constant, τ , the value of a first order system, when there is no input, is expressed by

$$x(t) = x(0)e^{-t/\tau} \tag{2}$$

where $x(0)$ is the initial value of the system. Similarly, the value of a system that starts at 0 and is given a step input u is expressed by

$$x(t) = Gu(1 - e^{-t/\tau}) \tag{3}$$

As a result, convenient way to model a first order system is to measure the system response to a step input and the gain at steady state input. An alternative and more useful way to model a system is to utilize the state-space method in which the system is described as:

$$\dot{x}(t) = A(t)x(t) + B(t)u(t) \tag{4}$$

$$y(t) = C(t)x(t) + D(t)u(t) \tag{5}$$

where x is the state vector, A is the system matrix, B is the input matrix, C is the output matrix, and D is the feed-forward matrix, y is the output vector and u is the input. Since it is assumed that the system is time invariant, the matrices can be written as $\{A, B, C, D\}$. If the formulation of the system is made in discrete domain instead of continuous domain, then the equations become

$$\begin{aligned}x(k+1) &= Ax(k) + Bu(k) & (6) \\y(k) &= Cx(k) + Du(k) & (7)\end{aligned}$$

Therefore, system identification is the process by which we identify the system matrices $\{A, B, C, D\}$. A Matlab program is generated to gather and plot the response of the robot wheels to step inputs and then the gain and time constant for each case is computed. Following these, the model is examined through another code and its output compared to actual data. Next, a program is developed which excites the robot's wheels with variety of speeds and using this data in Matlab script to generate a discrete state-space model of the system. Then, the response of the system model is plotted and compared with the actual response. An array is declared into which the data is stored during each iteration of the program's loop.

Locomotion Tests and Results

In the first test, by including necessary libraries: a state variable, which used to keep track of the state, and a duration variable, which used to progress through the state machine as desired time intervals, were initialized at 0. An initialization function was then called and the state was set to 1. A simple state machine was then used within the main function to perform a series of tasks in sequence in order to complete the desired path.

To easily change the speed parameter to any value, velocity mode was preferred. Each input to the velocity function was in mm/s. In the first state, the velocity for the left and right wheel were set to (250, 250) for 8 seconds to travel down the 2 m straight line. After 8 seconds of moving in a straight line in state 1, the state machine moved to state 2 where the velocities for the left and right wheel were set to (-250, 250) for 1.5 seconds to make the robot pivot to the left. The time needed to pivot was found experimentally. In the next state, the left and right wheel velocities were set to (400, 400) in order to observe the difference in speed when higher velocities were used. The robot should have taken approximately 5 seconds to return down the 2m track at 400mm/s, however only 3.8 seconds was allotted for the return trip so the robot did not collide with the wall. At the end of the 3.8 seconds, the robot was issued a velocity command of (0, 0) in order for it to stop.

Through consistent testing, the robot was able to navigate the path successfully however external forces such as carpet friction causes the robot to deviate from the path occasionally. To counter this, the robot was tested multiple times using the same program if it deviated to ensure that it was working successfully.

In the second test, after the initialization step, a 7-state state machine was used to navigate the robot around the track. In the first state, the velocity of the left and right wheels were set to (250, 250). Experimentally, it was found that 6.2 seconds were needed at 250 mm/s to navigate the first straight segment. This could be due to the carpet friction or the initial placement of the robot. For 0.7 m and 2.2 m straight segments (states 4, and 6) it theoretically should have taken 2.8 seconds and 8.8 seconds to complete respectively, however it experimentally took 3.1 seconds and 9 seconds to complete. This again could be due to carpet friction or the duration of the turn could have affected the length of the straight sections that the robot was required to navigate. To navigate the 0.4 m turn, Eq. 1 was used to calculate the required left and right wheel velocities needed. Thus, to navigate a 0.4 m radius turn to the right, the left wheel velocity has to be 1.28 times that of the right wheel velocity. This was confirmed experimentally by setting the left and right wheel velocities to (128, 100) to successfully complete 0.4 m radius turns. State 3 utilized the 0.4 m radius turn. Experimentally, it took an average of 5.51 seconds to complete 0.4 m radius turn at (128, 100) mm/s. The last state, state 7, commanded the robot to stop after the last turn.

In the last test, as in the previous tests, necessary libraries had to be included and an initialization step had to be executed to set the initial state as 1. After the initialization step, a 6-state state machine was used to navigate the robot around the track. The first state caused the robot to forward around the first half-circle. It was theoretically found that the left wheel velocity has to be 1.28 times that of the right wheel velocity. However, due to the surface conditions as well as other hardware related errors, it was experimentally found that when the left wheel velocity was 1.4 times that of the right wheel velocity, the robot successfully navigated around the 0.4 m radius track. As shown in Eq. 8, the robot should have taken approximately 11 seconds at left and right wheel velocities (140, 100) to traverse the half-circle of 0.4 m radius. However, experimentally it took approximately 11.4 seconds.

$$time = \frac{2\pi R}{\frac{(V_L + V_R)}{2}} \quad (8)$$

After 11.4 seconds passed, the program entered into the second state, which set wheel velocities to allow the robot to navigate to the left all the way around the second 0.4 m radius half-circle. According to the theoretical required velocity calculation, the right wheel velocity has to be 1.28 times that of the left wheel velocity. However, it was experimentally found that when the left wheel velocity is 1.5 times that of the right wheel velocity, the robot successfully navigated around the 0.4 m radius track. The robot should have taken approximately 11.7 seconds using left and right wheel velocities (100, 150) to traverse the half of the 0.4 m radius circle. Experimentally it took 12.2 seconds. This difference could be due to exactly when the robot began turning into the 0.4 m radius circle and the alignment of the robot over the track during the circle navigation. When 12.2 seconds passed and robot had finished navigating the second half-circle, the program entered into state 5 to navigate last line segment. The values had to be adjusted experimentally but stayed close to the theoretical values found. When the program entered in to state 5, the robot completed the last linear segment and then state 6 commanded the robot to stop.

System Identification Tests and Results

A program has been written to send step commands to move the wheels of the robot at the desired velocity. The outputs from the robot in all tests were collected at a frequency of 20 Hz. Considering the maximum speed, 500 mm/s, the robot can reach, it is obvious that this frequency is a quite suitable choice. A nested if-else loop was created to transfer files between the robot and the computer and execute robot movements depending on how much time has passed. For example; for the robot that was initially immobilized for 1 seconds, then moved with a velocity of 100 mm/s for three seconds and was commanded to stop again for another 1 seconds, the counter was started at 0 and increased 20 times every second when the program looped back through. For this process, the inputs to the wheels, the output which is the velocity of the both wheels resulting from the input, and odometer readings were written in the matrix with a frequency of 20 Hz. A counter was utilized to increment through the rows of the data file and record the required information. After all locomotion stages were executed and the necessary data was written to the matrix, the data file was sent back to the computer via ftp server.

Once the code was written for the robot to execute each of the desired movements and return the data back to a file on the computer, plots had to be generated using the data in order for the data to be analyzed. Each column in the file was then assigned to a variable which represented each of the required data arrays. Once each variable was assigned to a data array, the input, left wheel velocity, and right wheel velocity were plotted. The following Table 1 lists the performed tests and the results. For all tests, the gain and time constant for the step input from 0 to the desired level and for the step input from the desired level to 0 had to be estimated. Computed gain and time constant values are given in the table below.

Table 1:- Results of the system identification tests.

Test	Mode	$u(k)$	G	$\tau \leftarrow 0$	$\tau \rightarrow 0$
1	V-L	0.1 m/s	1.079	0.211	0.139
2	V-L	0.2 m/s	1.061	0.227	0.194
3	V-L	0.3 m/s	1.063	0.258	0.173
4	V-L	0.4 m/s	1.055	0.231	0.125
5	V-R	0.1 m/s	1.033	0.122	0.137
6	V-R	0.2 m/s	1.027	0.139	0.111
7	V-R	0.3 m/s	1.019	0.145	0.203
8	V-R	0.4 m/s	1.021	0.188	0.174

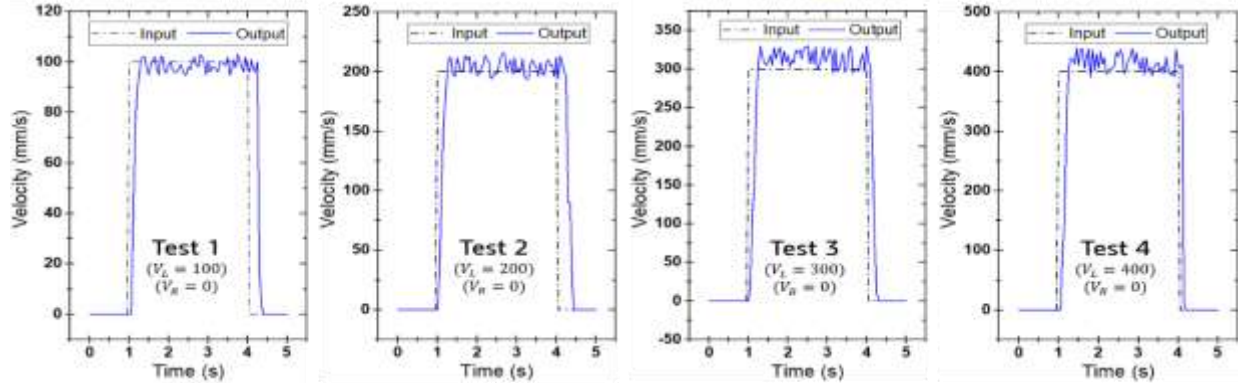


Figure 5:- Responses of the left wheel of the robot to various input values.

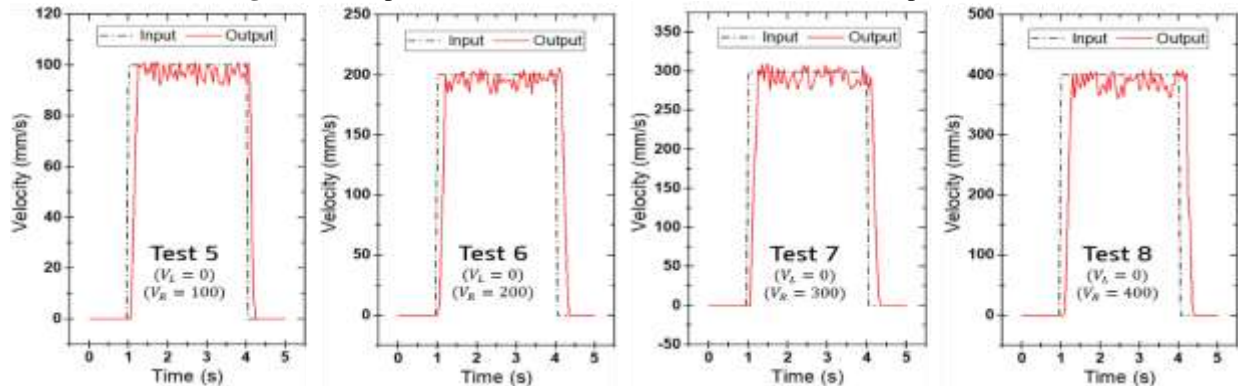


Figure 6:- Responses of the right wheel of the robot to various input values.

In the table above, mode means the following: (V-L): Velocity mode, left wheel only. The robot will pivot about the other wheel; (V-R): Velocity mode, right wheel only. The robot will pivot about the other wheel; The figures obtained from plotting the input and actual robot wheel velocities from tests 1-8 are shown above in Fig. 5 and Fig. 6.

The plots depict the results of each test individually. When the response of the left wheel of the robot to the system input is examined, a velocity variation is observed around the expected value at low velocities ($v_i = 100$ and $v_i = 200$), but a velocity variation above the expected value is observed at higher velocities ($v_i = 300$ and $v_i = 400$). In addition, output signals have a high amount of noise for all cases as can be clearly seen from the graphs. Interestingly, for the velocity profile from desired level to 0; relatively distinct velocity values from the step input were obtained for low velocity values. Conversely, velocity values close to the step input were obtained for high speed values.

According to the response of the right wheel of the robot to the system input, a velocity variation is observed slightly under the expected value for all velocities. Similar to the previous cases, output signals have a high amount of noise for all cases as can be clearly seen from the graphs. In all four cases, the velocity profiles from 0 to desired level and the velocity profiles from desired level to 0 show similar patterns. From these figures, it can be concluded that the velocity output of the right wheel produces more stable output.

Fig. 7 displays the normalized (indicated by asterisk, ‘*’) outputs for each case. From the figures, it can be observed that the step input function provides more stable output at higher input velocities. It is noticed that the test results for left wheel are fairly consistent when reaching the desired velocity level from 0, but differ when returning from the desired level to 0. This indicates that the left wheel of the robot is sensitive to the input velocity value. This is most likely due to left servo motor or hardware related issue that needs to be resolved. When the same alteration is examined for the right wheel, it is obvious that the results differ slightly from the test results for the left wheel. The results are more consistent compared to the left wheel, both when reaching from 0 to the desired velocity level and returning from that level to 0.

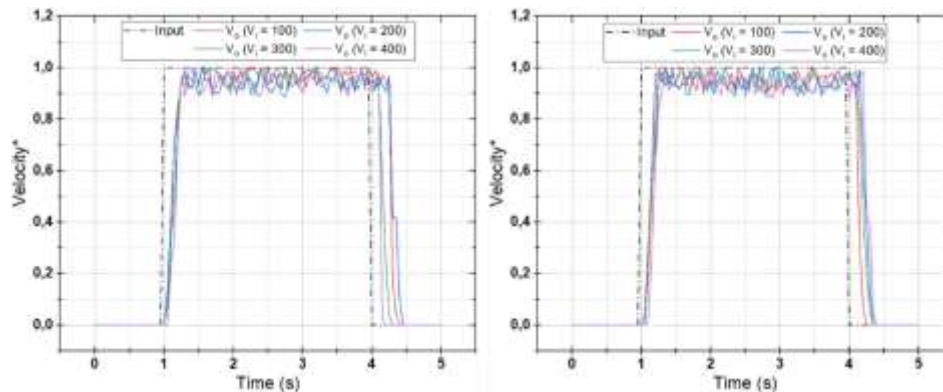


Figure 7:- Dimensionless* (normalized) velocity outputs for left (left) and right (right) wheels of the robot.

Conclusion:-

In this study, the velocity propulsion mode of a servo motor robot was investigated. With this review, a preliminary test was performed prior to the system identification study for the hardware analysis of the robot. In the first of the tests performed in three stages, the simple linear motion of the robot and its rotation around its own axis were examined. In the second and third tests, relatively more complex routes were predetermined and the accuracy of the robot to complete these routes was tested. According to the test results, it has been observed that the robot successfully completed these routes with minor errors. The difference between theoretical and experimental findings may arise from environmental conditions such as floor conditions as well as from the hardware of the robot. Considering the materials from which the robot is produced, it can be concluded that most of the error is caused by the robot itself. To make a mobile robot perform the desired movements and system identification, a data-driven model for the robotic platform was developed. For the wheels of the robot, various system identification tests have been conducted to determine the gain of each steady state level and time constant for entering the desired velocity level from 0 and returning from the desired level to 0. The responses of both wheels of the robot to the specific step input functions were tested. In the first part of the tests, the right wheel of the robot was left immobilized and only the left wheel was commanded to move with specific velocities, allowing the robot to rotate around the right wheel. In the next part, this time the left wheel of the robot was left stationary, and only the right wheel was sent a velocity command to turn the other wheel around. The responses of both wheels of the robot to the input command were examined and gain and time constant values were computed for each test. According to the test results, it was observed that the right wheel of the robot was more consistent and produced closer to the expected value for each test performed. Moreover, it was concluded that there was probably a hardware error by examining the different velocity profiles displayed when the left wheel returned from the desired level to 0 in each test.

References:-

1. Alves, T.G., Lages, W.F., Henriques, R.V.B. (2018). Parametric identification and controller design for a differential drive mobile robot. *Int. Fed. Of Auto. Control*, 51(15), 437-442.
2. Berceanu, C. and Tarnita, D. (2012). Mechanical design and control issues of adexterous robotic hand. *Advanced Materials Research*, 463, 1268-1271.
3. Bruder, D., Remy, C.D., Vasudevan, R. (2019). Nonlinear system identification of soft robot dynamics using Koopman operator theory. *IEEE International Conference on Robotics and Automation*, Montreal, Canada, 1-7.
4. Chen, K.Y., Lai, Y.H., Fung, R.F. (2017). A comparison of fitness functions for identifying an LCD glass-handling robot system. *Mechatronics*, 46, 126-142.
5. Corrales, A., Malfaz, M., Salichs, M. (2014). Signage system for the navigation of autonomous robots in indoor environments. *IEEE Transactions on Industrial Informatics*, 10(1), 1-9.
6. Gonzales, A.G.C., Alves, M.V.S., Viana, G.S., Carvalho, L.K., Basilio, J.C. (2018). Supervisory control-based navigation architecture: a new framework for autonomous robots in industry 4.0 environments. *IEEE Trans. On Ind. Informatics*, 14(4), 1732-1743.
7. Howell, A., Way, E., McGrann, R., Woods, R. (2006). Autonomous robots as a generic teaching tool. *36th ASEE/IEEE Frontiers in Education Conference*, San Diego, CA, 17-21.
8. Malu S.K., Majumdar, J. (2014). Kinematics, localization and control of differential drive mobile robot. *Glb. J. of Res. in Eng.: H Rob. & Nanotech*, 14(1), 1-7.

9. Mitnik, R., Nussbaum, M., Soto, A. (2008). An autonomous educational mobile robot mediator. *Auton Robot*, 25, 367-382.
10. Said, A., Davizon, Y., Soto, R., Herran, C.F., Santos, C.H., Roman, P.E. (2018). An infinite-norm algorithm for joystick kinematic control of two-wheeled vehicles. *Electronics*, 7(164), 1-15.
11. Sasiadek, J. (2014). Space robotics - Present and past challenges. *19th International Conference on Methods and Models in Automation and Robotics*, MMAR 2014. 926-929. 10.1109/MMAR.2014.6957481.
12. Sheridan, T. (2016). Human-robot interaction: status and challenges. *Human Factors*. 58. 10.1177/0018720816644364.
13. Nikolaidis, S., Nath, S., Procaccia, A., Srinivasa, S. (2017). Game-theoretic modeling of human adaptation in human-robot collaboration. *IEEE International Conference on Human-Robot Interaction*, Vienna, Austria, 323-331.
14. Decker, M., Fischer, M., Ott, I. (2017). Service robotics and human labor: a first technology assessment of substitution and cooperation. *Robotics and Autonomous Systems*, 87, 348-354.
15. Carsten, J., Rankin, A., Ferguson, D. (2007). Global path planning on board the mars exploration rovers. *IEEE Aerospace Conference*, 1-11, DOI: 10.1109/AERO.2007.352683.

16. Lingshuai, M., Yang, L., Haitao, G. (2016). A new type of small underwater robot for small scale ocean observation. *6th Annual IEEE International Conference on Cyber Technology in Automation, Control and Intelligent Systems*, Chengdu, China, 19-22.
17. Rubio, F., Valero, F., Llopis-Albert, C. (2019). A review of mobile robots: concepts, methods, theoretical framework, and applications. *International Journal of Advanced Robotic Systems*, 1-22, DOI: 10.1177/1729881419839596.
18. Bevacqua, G., Cacace, J., Finzi, A. (2015). Mixed-initiative planning and execution for multiple drones in search and rescue missions. *25th International Conference on Automated Planning and Scheduling*, Jerusalem, 315-323.
19. Kim, S.K., Tilbury, D.M. (2004). Mathematical modeling and experimental identification of an unmanned helicopter robot with flybar dynamics. *J. of Robotic Systems*, 21(3), 95-116.
20. Schafroth, D., Bermes, C., Bouabdallah, S., Siegwart, R. (2010). Modeling, system identification and robust control of a coaxial micro helicopter. *Control Engineering Practise*, 18(7), 700-711.
21. Soderstrom, T. (2010). System identification for the errors-in-variables problem. *UKACC International Conference on Control*, Coventry, 19-32.
22. Wang, H. (2013). Task-space synchronization of networked robotic systems with uncertain kinematics and dynamics. *IEEE Transactions on Automatic Control*, 58(12), 3169-3174.
23. Xu, J.X. (2011). A survey on iterative learning control for nonlinear systems. *International Journal of Control*, 84(7), 1275-1294.
24. Chung, S.J., Slotine, J.J.E. (2009). Cooperative robot control and concurrent synchronization of Lagrangian systems. *IEEE Transactions on Robotics*, 25(3), 686-700.
25. Gupta, A., Murali, A., Gandhi, D., Pinto, L. (2018). Robot learning in homes: improving generalization and reducing dataset bias. ArXiv, abs/1807.07049, 1-10.
26. Ulmen, J., Cutkosky, M. (2010). A robust, low-cost and low-noise artificial skin for human-friendly robots. *IEEE International Conference on Robotics and Automation*, Anchorage, Alaska, USA, 4836-4841.
27. Ouyang, P.R., Acob, J., Pano, V. (2014). PD with sliding mode control for trajectory tracking of robotic system. *Robotics and Computer-Integrated Manufacturing*, 30(2), 189-200.
28. Liu, Y.C., Chopra, N. (2010). Synchronization of networked robotic systems on strongly connected graphs. *49th IEEE Conference on Decision and Control*, Atlanta, Georgia, USA, 3194-3199.
29. Yannier, S., Sabanovic, A. (2012). Obstacle avoiding and goal point tracking layered structure for mobile robots. *Conference on Electrical and Electronics Engineering*, 1-6, Bursa, Turkey.
30. Yilmaz, N., Sagiroglu, S., and Bayrak, M. (2006). General aimed web based mobile robot: Sunar. *Gazi University Journal of Faculty of Engineering and Architecture*, 21(4), 745-752.