Underactuated Two-Wheeled Mobile Robot

Hardik Gossain

Department of Electrical and Computer Engineering, University of Waterloo, 200 University Ave. W, Waterloo, Ontario, N2L 3G1, Canada

hgossain@uwaterloo.ca

***Abstract*— The implementation and testing of real-time control of a two-wheeled mobile robot are described in this study. The mobile robot used in this study is underactuated with three inputs for mobility and two input for control. The challenge is to control the oscillation of the intermediate body while following the path. The robot is tested in three motions: translational, rotational, and uphill. A linear-quadratic regulator is designed and tested with the help of the simulation and required parameters are compared to get the best result**

***Keywords*— underactuated, mobility, oscillation, linear-quadratic**

I. Background

Two-wheeled robots are gaining popularity due to their simple mechanical design. This paper is all about QUASIMORO (Quasiholonomic Mobile Robot), which was designed and built at McGill University's Centre for Intelligent Machines. There has been a lot of research done on 2 wheeled robots. As it is a system with two wheels parallel to each other with an intermediate body in between them, there is always a challenge to stabilize it in equilibrium. There are stabilization methods used to stabilize it. Passive stabilization is where the robot is stabilized mechanically, generally by using a third wheel (castor wheel) and another is active stabilization. The mass centre of the robot plays the main role in active stabilization which makes the robot unstable at its equilibrium state. To simplify control of the robot, the mass was placed below the wheel axis as shown in Fig 1.

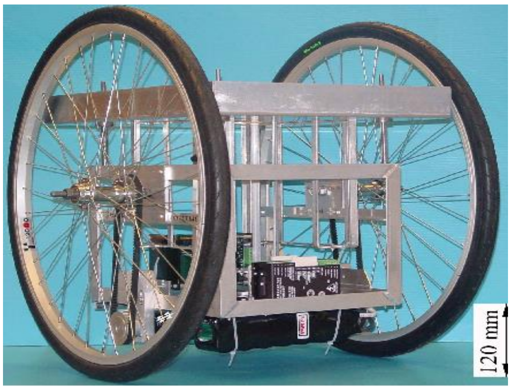

Fig. 1 Quasimoro prototype

There are many other types of wheeled robots with either less payload or with a third wheel but here significant payload almost one-third of the robot's total mass was used.

The robot used a state feedback controller. As the robot doesn't have any third wheel, hence the robot is underactuated which means the robot would have to deal with two problems i.e., firstly stabilizing the intermediate body and also following the desired path. The robot focuses to settles the intermediate body at its equilibrium point. This simply simplifies the robot control which enabled the use of linear state feedback control.

While working on this two-wheeled mobile robot the primary focus for our research is the feedback control for the robot. Quasimoro controls its IB to settle at a stable equilibrium point. This significantly simplifies the dynamics of the robot, thereby allowing the use of a simple linear state-feedback control law.

II. Model of the Robot

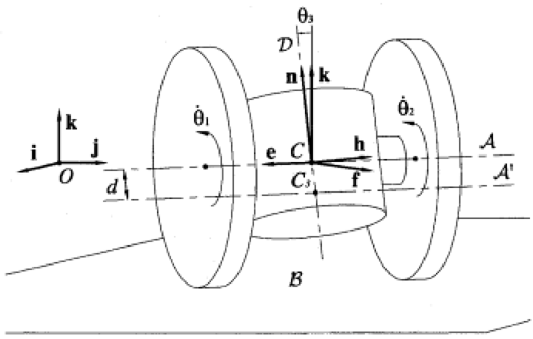

Fig. 2 mathematical model of robot

The above robot is modelled to state space form. The parameters of the system used are:

J_1= 0.591 kg m^2
J_2= 0.628 kg m^2
d= 0.120 m
m_{wheels}= 3.459kg
m_{body}= 16.222kg
r_{wheel}= 0.295m
l= 0.480m

State Equations are

$$\dot{x} = Ax + Bu$$
$$y = Cx$$

It is further linearized at its equilibrium position keeping $x_0=0$.

$$A \equiv \frac{\partial K}{\partial x} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 3.452 & 0 & 0 & 0 \\ 0 & 0 & 3.452 & 0 & 0 & 0 \\ 0 & 0 & -20.720 & 0 & 0 & 0 \end{bmatrix}$$

$$B \equiv \frac{\partial k}{\partial u} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0.892 & -0.049 \\ -0.049 & 0.892 \\ -0.238 & -0.238 \end{bmatrix}$$

Here x is the state vector with the following state variables

$$x = [\theta_1\ \theta_2\ \theta_3\ \dot{\theta}_1\ \dot{\theta}_2\ \dot{\theta}_3]$$

U_1 and U_2 are the inputs for two wheels representing there torques

The A matrix is not full ranked and hence the system will be not controllable. Also, the desired output cannot be tracked by full state feedback. To solve this problem, states of the model can be reducing to the following:

$$x_r = [\theta_3\ \dot{\theta}_1\ \dot{\theta}_2\ \dot{\theta}_3]$$

Further partial state feedback control law is used:

$$u = My - Kx_r$$

here x_r is the reduced state vector. M and K are the gain matrices for the input and the state feedback respectively.
Hence the reduced state model of the system is as follows

$$x_r = [\theta_3\ \dot{\theta}_1\ \dot{\theta}_2\ \dot{\theta}_3]$$

$$A_r = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 3.452 & 0 & 0 & 0 \\ 3.452 & 0 & 0 & 0 \\ -20.720 & 0 & 0 & 0 \end{bmatrix}$$

$$B_r = \begin{bmatrix} 0 & 0 \\ 0.892 & -0.049 \\ -0.049 & 0.892 \\ -0.238 & -0.238 \end{bmatrix}$$

$$C_r = \begin{bmatrix} 0 & 0.1475 & 0.1475 & 0 \\ 0 & -0.6150 & 0.6150 & 0 \end{bmatrix}$$

The controller designed for the system uses Linear Quadratic Regulator. This is a widely used control design. It provides a optimally controlled feedback gain that is a K_r matrix that minimizes the integral (i.e. the energy used).

$$J = \int_0^{\infty} [\, x_r^T(t) Q_r x_r(t) + u^T(t) R u(t)]\, dt$$

where Q and R are symmetric, and positive semidefinite positive-definite weighting matrices, respectively.
The gain matrix K_r is computed from the following equation:

$$K_r = R^{-1} B_r^T S$$

Where S is unique, symmetric and positive semidefinite and the solution of the Riccati Equation

$$SA_r + A_r^T S + Q - SBR^{-1}B^T S = 0$$

Here 0 is not scalar but can be said a 4x4 matrix of zeros.

$$\mathbf{K}_r = \begin{bmatrix} -3.2 & 0.65 & 0 & -4.5 \\ -3.2 & 0 & 0.65 & -4.5 \end{bmatrix}$$

$$\mathbf{M}_r = \begin{bmatrix} 2.1 & -0.47 \\ 2.1 & 0.47 \end{bmatrix}$$

III. Simulation

The above model was the originally made with reference to actual robot that was built at McGill University, Canada. Hence to create my own simulation, the factors like the dimensions and mass of the original model were used to make an approximate duplicate model of the original in the software. MATLAB and Simulink were used to perform the simulation of the model. The original model uses two inputs, one input for each wheel to steer the robot while to simplify the simulation, reference to an inverted pendulum cart or self-balancing robot was used.

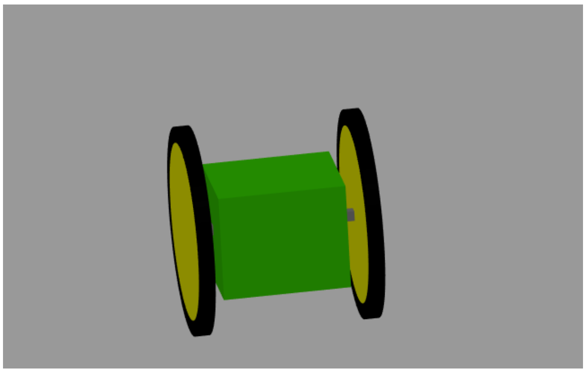

Fig. 3 simulated model of mobile robot

$$\dot{x} = Ax + Bu$$
$$y = Cx$$

The parameter used are:

$J_1 = 0.5$ kg m^2
$J_2 = 0.628$ kg m^2
$d = 0.120$ m
$m_{wheels} = 3.5$ kg
$m_{body} = 16$ kg
$r_{wheel} = 0.295$m

$$A=\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -0.0053 & 0.4005 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & -0.0058 & 11.1568 & 0 \end{bmatrix}$$

$$B=\begin{bmatrix} 0 \\ 0.0534 \\ 0 \\ 0.0534 \end{bmatrix}$$

$$C=\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

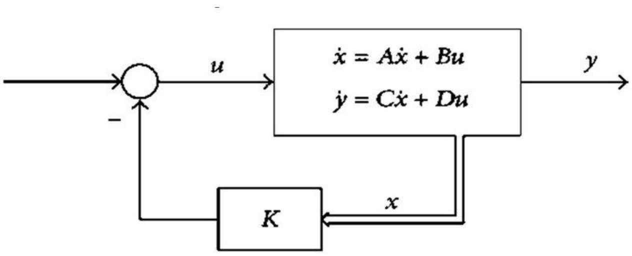

Fig. 4 LQR controller block diagram

As mentioned above, LQR is applied to get an optimal controlled feedback gain K.

Matrix Q and R were chosen as standard were,

$$Q=C^TC$$

$$R=1$$

With these values of Q and R, the gain matrix K was calculated.

$$K=\begin{bmatrix} -0.0791 & -1.9084 & 392.6758 & 117.6254 \end{bmatrix}$$

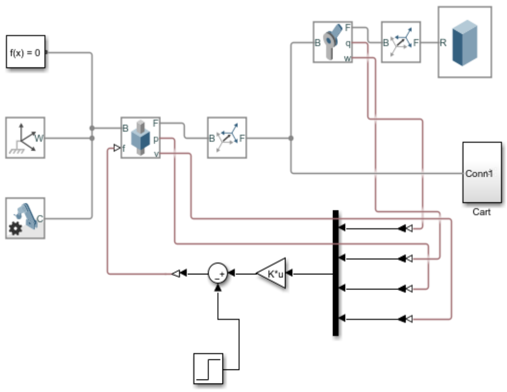

Fig.5 Simulink model of our simulation

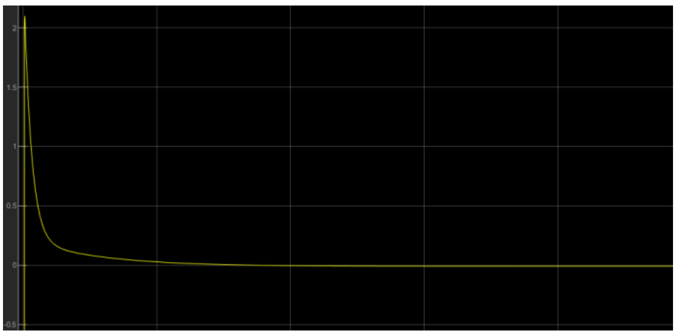

Fig. 6 stability graph obtained of mobile-robot

The above graph describes the stabilization to the robot to its equilibrium state using the LQR control.

In order to check the efficiency of the controller I applied PID control (Proportional Integral Derivative) to check weather LQR is the best option or PID could give better results. This is the result of PID controller.

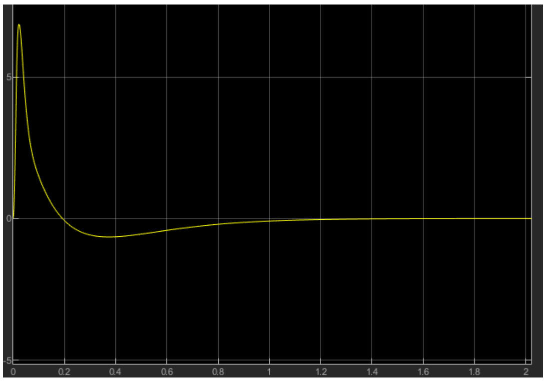

Fig. 7 Stabilization graph using PID control

By observing the result of this controller, it can be said that LQR control works better. The main reason is that, for every different scenario the constants values for the PID controller has to be changed while LQR doesn't require such changes and hence for this model LQR works efficiently well.

The original paper used the original robot to perform various tests and observe its behaviour in real physical conditions. It was difficult to simulate these conditions in MATLAB hence it was just studied and analysed. These are the list of such conditions in which the physical robot was tested.

A. Robot Translation and Payload Variation

Analysis in the reference article was done to determine the changes caused due to payload variation. 70 N is the maximum nominal payload. The load utilized in this example is about 15 N. The fixed-distance test was chosen, with zero to four 15-N weights put on the robot tray. As a result, the 1.5-m motion was repeated five times with a different payload each time. There was no discernible difference between the tests based on visual observation. Though after 60N there was a slightly larger variation in overshoot.

B. Reaching a Stationary Orientation

The goal of the test was to achieve a 180-degree angular displacement with no translational motion. 3 rad/s is the maximum speed obtained. There is a little overshoot in the data, showing that the robot settles at an angle of 3.158 rad rather than 3.1416 rad. This minor inaccuracy, together with the error associated with the manual measurement of the distances with which the angular displacement was computed, is again attributed to slippage.

C. Robot Rotation and Payload variation

Because no reactive forces or moments happen on the IB during the robot's rotation, the payload should have no influence on the tilt. To keep the same velocity, increasing the payload should merely demand a larger torque from the motors, which should not be an issue. to confirm We next performed the rotation test while changing the payload. The payload, as planned, had no visible payload. During a rotation, this has an impact on performance. The speed is easily noticeable. The maximum tilt values, which fluctuate, are managed and maintained. between 1.169 and 1.283
, rather than being caused by the tilt-sensor noise, the motion itself.

D. Straight Path Then Turn

The IB moves 1.5 meters forward, turns 180 degrees, and then moves 1.5 meters forward again to return to its original position. The robot completes the action, however, there is a tiny variation between the beginning and end positions, according to examination. We can observe that the robot does not return to its exact beginning location since it turns slightly over 180 degrees. We can also see an overshoot, which is caused by the IB swinging during deceleration, but the robot corrects it in under 3 seconds. The placement error is less than 20 millimetres.

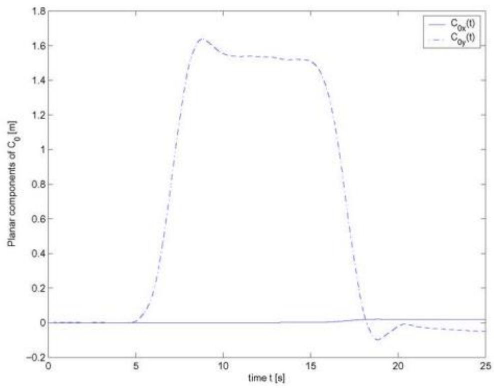

Fig. 8 straight and turn motion

E. Arbitrary Velocities

Using the joystick in manual control mode, the robot travels through a series of arbitrary velocities. Quasimoro is assigned a history of random velocities, and we additionally record the observed tilt 3 of the IB, which rises by just 0.1 rad.

F. Sudden Stop

Quasimoro's reaction to an abrupt stop was studied. The joystick trigger button is used to bring the robot to a complete halt after 3 seconds. The IB shifted forward as a result of this. Because of the IB's swing, velocity increased for a fraction of a second before dropping to zero in less than 2 seconds. In less than 3 seconds, IB stability was achieved. During this movement, the highest tilt observed was close to 0.2 rad.

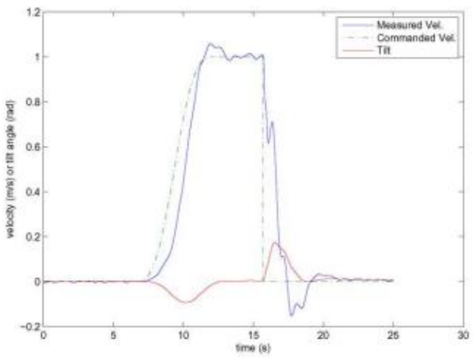

Fig. 9 Graph for velocity and tilt angle for sudden stop

G. Uphill Motion

The robot was able to effectively climb a 20% incline while maintaining the appropriate pace and keeping the IB horizontal rather than parallel to the ramp during an outdoor test. During an indoor test, the robot was also able to effectively climb a 10% incline. During the test, the robot accelerates to 1 m/s in 2 seconds, then maintains a constant speed of 1 m/s over the incline, then decelerates till it comes to a complete stop in 2 seconds.
The 10 percent elevation was achieved by placing one flat 1.5 m table on the left, an inclined 1.5 m table in the middle, and then another flat 1.5 m table on the right.

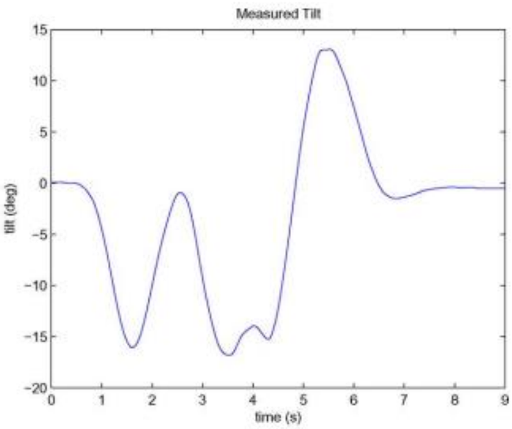

Fig. 10 Graph for tilt angle on uphill motion

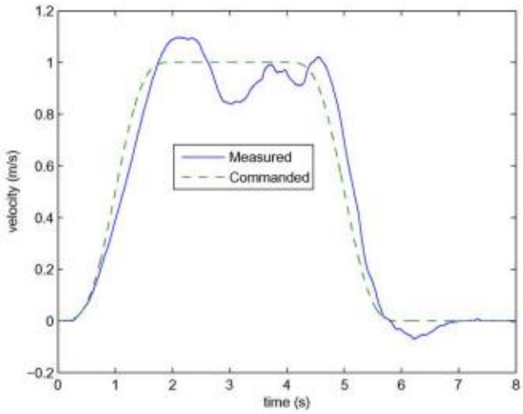

Fig. 11 Graph for velocities on uphill motion

IV. Conclusion

It can be concluded that the research paper was very well written and performed. All the graphs and the experiments performed were all tested on a real robot so that values of gains mentioned in the paper were practical. In order to analyse the paper and verify that the model worked perfectly, I applied PID controller to check weather it could work better on the model or not, which turned out to be false and concluded that LQR control worked best for such kind of model.

References

[1] P. Oryschuk, A. Salerno, A. M. Al-Husseini and J. Angeles, "Experimental Validation of an Underactuated Two-Wheeled Mobile Robot," in IEEE/ASME Transactions on Mechatronics, vol. 14, no. 2, pp. 252-257, April 2009, doi: 10.1109/TMECH.2008.2007482.

[2] A. Salerno and J. Angeles, "The control of semi-autonomous two-wheeled robots undergoing large payload-variations," IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04. 2004, 2004, pp. 1740-1745 Vol.2, doi: 10.1109/ROBOT.2004.1308075.

[3] Christopher Nielsen. ECE 682: Multivariable Control Systems Lecture Notes. University of Waterloo, Fall 2021.

[4] A. Salerno and J. Angeles, “The control of semi-autonomous two wheeled robots undergoing large payload-variations,” in Proc. IEEE Int. Conf. Robot. Autom., New Orleans, LA, 2004, pp. 1740–1745.

[5] A. Salerno and J. Angeles, “On the nonlinear controllability of a quasiholonomic mobile robot,” in Proc. IEEE Int. Conf. Robot. Autom., Taipei, Taiwan, Sep. 2003, pp. 3379–3384.