

Linear and Non-linear Control Design of Skid Steer Mobile Robot on an Embedded Board

Jharna Majumdar, Sudip C Gupta, B Prassanna Prasath

Centre for Robotics Research, Nitte Meenakshi Institute of Technology, P. B. No. 6429, Yelahanka, Bangalore, India

Article Info

Article history:

Received Feb 12, 2018
Revised Jul 18, 2018
Accepted Aug 13, 2018

Keyword:

LQR
Maxon RE40
Mbed LPC 1768
PID
SSMR

ABSTRACT

A detailed approach for a linear Proportional-Integral-Derivative (PID) controller and a non-linear controller-Linear Quadratic Regulator (LQR) is discussed in this paper. By analyzing several mathematical designs for the Skid Steer Mobile Robot (SSMR), the controllers are implemented in an embedded microcontroller-Mbed LPC1768. To verify the controllers, Matlab-Simulink is used for the simulation of both the controllers involving motors-Maxon RE40. This paper compares between PID and LQR controller along with the performance comparison between Homogenous and Non-Homogenous LQR controllers.

Copyright © 2018 Institute of Advanced Engineering and Science.
All rights reserved.

Corresponding Author:

Jharna Majumdar,
Centre for Robotics Research,
Nitte Meenakshi Institute of Technology,
P.B.No.6429, Yelahanka, Bangalore, India.
Email: jharna.majumdar@nmit.ac.in¹, sidishere.sudip@gmail.com², prassanna4robo@gmail.com³

1. INTRODUCTION

The interest in the domain of mobile robots and its automation has grown in the past few years. There are lots of research on the kinematics and dynamic modelling, and for the control system for the mobilization of the robot, which should satisfy its performance based on time and accuracy. Although several research were conducted on the control system [1], the implementation on an embedded board has not been mentioned involving memory management and low redundancy. There were several research related to various approaches to achieve the optimum control system and their respective simulation results proved to be optimum [1-6], but when used in a system or a robot, the output determines all these models to be sparse.

Skid Steer Mobile Robots (SSMR) are the vehicles that can tract on all terrain condition. To move the SSMR on all terrain there must be a feedback determining the errors in its predicted parameters and the output parameters, which must be done autonomously by the embedded controller, which provides the actuation signals to all the wheels in terms of pulse-width modulation (PWM). This can be achieved by implementing the two control algorithms (PID and LQR) in the embedded controller Mbed LPC1768.

2. SSMR MODELING

The Skid Steering Mobile Robot is a four-wheeled high traction mobile robot, which are steered by differential drive. A strong controller is required to compute all the values of the mobile robot and hence Mbed LPC1768 is preferred taking cost effectiveness and speed of operation into consideration. The SSMR Model is observed through 3D CAD CATIA is shown in Figure 1.

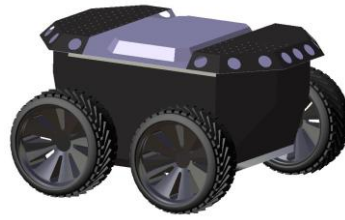


Figure 1. CATIA model of SSMR

A mathematical description of the dynamics of an SSMR moving on a planar surface is introduced in this section. The mathematical model of the vehicle can be divided into three parts: kinematics, dynamics and drive subsystems. Here, we focus on the first two blocks, i.e., the drive and dynamics subsystems, and we use them for reference tracking control of both the linear and angular velocities. SSMR model as shown in Figure 2.

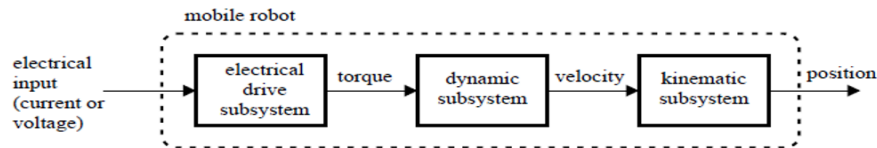


Figure 2. SSMR Model

2.1. Dynamic modelling

This process allowed a thorough study for the design and the load carrying capacity of the SSMR and the optimization in its weight, which is very beneficial for its control. In Figure 3 is schematic model of SSMR in global coordinate system. The main equation that describes the dynamic subsystem of the SSMR moving on a planar surface as shown in figure is given by (1):

$$M(q) \ddot{\eta} + C(q) \dot{\eta} + R(q) = B(q) \tau \tag{1}$$

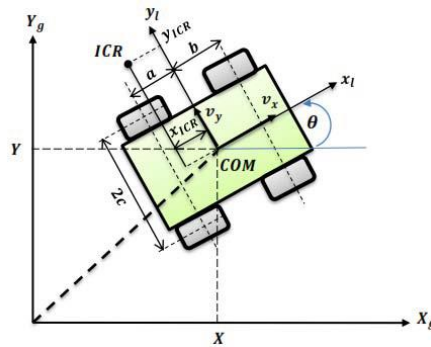


Figure 3. Schematic Model of SSMR in Global Coordinate System

2.2. Drive model

In Figure 4 is drive model. Four DC motors coupled with mechanical gears finally to the wheels drive the SSMR. Considering only one of the drives (Motor) and assuming remaining all motors to have same parameters, the relation between torque τ and voltage u_{vd} can be written as shown in (2-3).

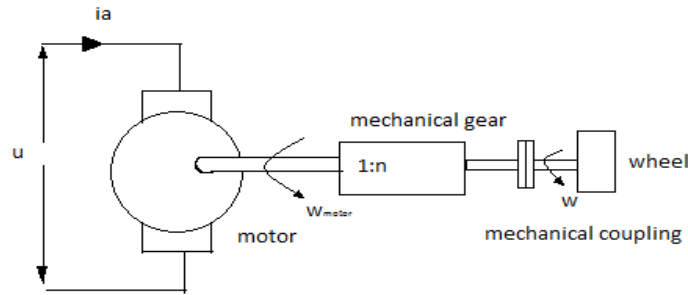


Figure 4. Drive Model

$$\tau = nK_i i_a \tag{2}$$

$$u_{va} = L_a \frac{d}{dt}(i_a) + R_a i_a + nK_e w_i \tag{3}$$

Where,

i_a is the armature current,

K_i is the motor torque constant,

N is the gear ratio ($n > 1$),

L_a and R_a denote the series inductance and resistance of the rotors respectively,

K_e is the electromotive force coefficient, and

$W = [w_1 \ w_r] \ T$.

The left w_l and right w_r sides angular velocities can be obtained from the following formulas as shown in (4-5)

$$w_l = \frac{vx - \text{present error}}{r} \tag{4}$$

$$w_r = \frac{vx - \text{present error}}{r} \tag{5}$$

The Matlab modelling of the drive is shown below in the Figure 5

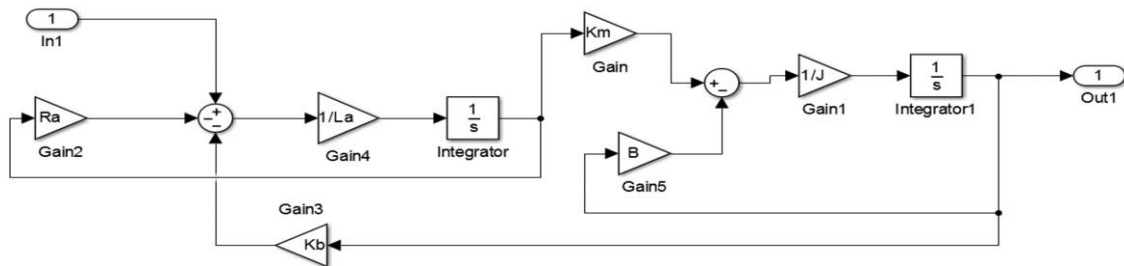


Figure. 5 Drive model in Matlab

3. CONTROLLER ALGORITHM

3.1. PID controller

PID is the basic linear control algorithm used in a system, which involved a constant feedback gain fed to the input. A PID controller determines error values as a difference between required value and measure process variables. PID involves three parameters i.e. Proportional (P), Integral (I) and Derivative (D) where P is accountable for present values, I for past values and D for future values. Block diagram of PID controller as shown in Figure 6.

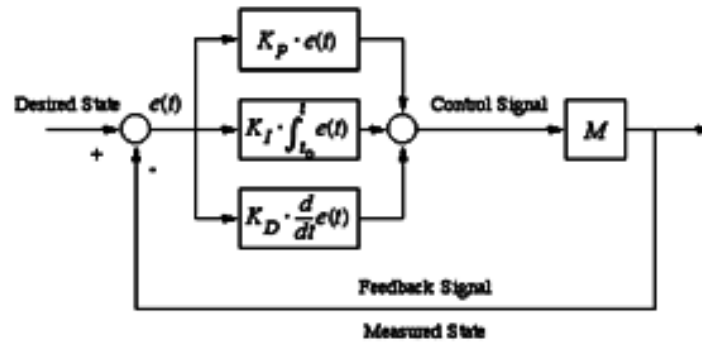


Figure 6. Block diagram of PID controller

The basic governing equation of PID controller is given by (6):

$$u(t) = K_p e(t) + K_i \int e(t) dt + K_d \frac{de}{dt} \tag{6}$$

The values of P, I, D were manually tuned to obtain the best response with the help of matlabsimulation, which was then fed into the actual controller design for the SSMR as C program. Matlab simulation model of the controller design and its response can be observed in the Figure 7 and 8 below where PID controller is attached to drive model, which is derived earlier.

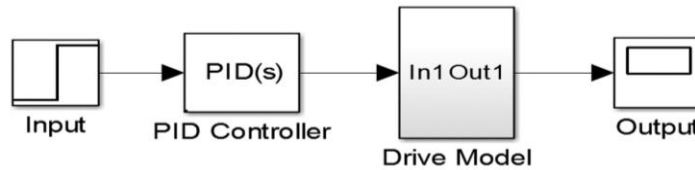


Figure 7. Flow Diagram of PID Controller

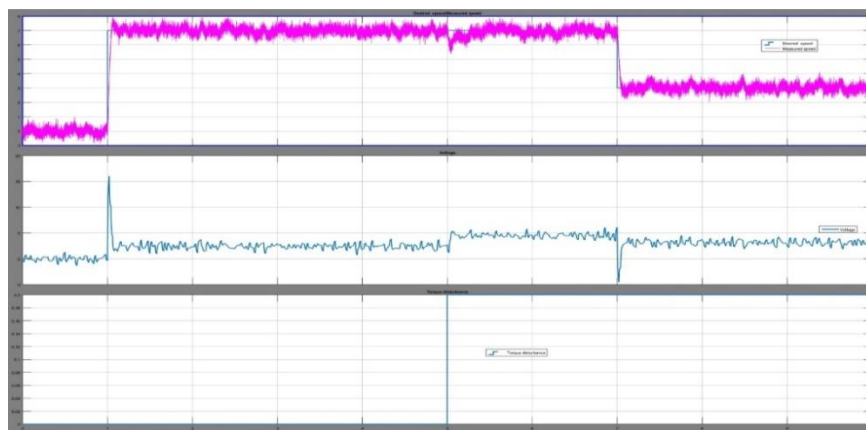


Figure 8. Response of PID Controller

3.1.1. Implementation of LQR on NXP LPC 1768 microcontroller

LPC 1768 controller has a 32-bit ARM Cortex-M3 core running at 96MHz. It also has 512KB flash, 32KB RAM and lots of I/O pins, which are required for the operation of SSMR. Pin diagram of Mbed LPC1768 as shown in Figure 9.

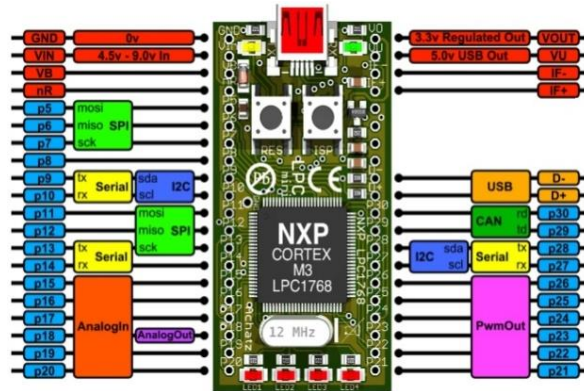


Figure 9. Pin Diagram of Mbed LPC1768

Algorithm

- Step 1** Initialize the pins of the microcontroller.
- Step 2** Define parameters like gains, K, P, I, D
- Step 3** Input the desired speed
- Step 4** Calculate the count per second “t” for the desired speed in one revolution
- Step 5** Assign interrupts for the counts
- Step 6** Receive the feedback from the encoder for time “t”
- Step 7** Compare the desired speed and feedback for time “t”
- Step 8** Calculate the error using the PID general formula
- Step 9** compare the error with the set gain
- Step 10** Generate PWM using respective gains and errors as base PWM
- Step 11** Run the entire step in loop.

3.2. LQR controller

Although the PID controller worked on the SSMR and showed an acceptable mobility, it was flawed. The idea of linear feedback proved to make the SSMR unable to move under varying terrain conditions. To overcome this problem, LQR controller was implemented on the SSMR. LQR is an optimal controller as it provides the smallest possible error in its output when compared to the input. The later controller design was optimally in paper [1]. By taking all the parameters needed for feedback model, drive model and dynamic model as derived in paper [1][2] further analysis were imbued in the Mbed LPC1768 microcontroller. The block diagram of LQR controller as shown in Figure 10.

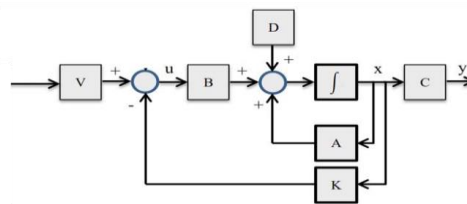


Figure 10. Block Diagram of LQR Controller

3.2.1. Dynamic model

The dynamic model consists of the state space evaluation of M, C, R, and B are formulated (7-10)

$$\bar{M} = \begin{bmatrix} m & 0 \\ 0 & mx_{ICR}^2 \end{bmatrix} \tag{7}$$

$$\bar{C} = \begin{bmatrix} 0 & mx_{ICR} \dot{\theta} \\ -mx_{ICR} \dot{\theta} & mx_{ICR} \dot{x}_{ICR} \end{bmatrix} \tag{8}$$

$$\bar{R} = \begin{bmatrix} F_{rx}(\dot{q}) \\ x_{ICR} F_{ry}(\dot{q}) + M_r \end{bmatrix} \tag{9}$$

$$\bar{B} = \begin{bmatrix} \frac{1}{r} & \frac{1}{r} \\ -\frac{c}{r} & \frac{c}{r} \end{bmatrix} \quad (10)$$

Here, m is the mass of the body, θ is the orientation of the robot with respect to the global coordinates. Using all these values, we substitute them in the state equation (11).

$$\dot{\eta} = \bar{M}^{-1}(q)\bar{B}(q)\tau - \bar{M}^{-1}(q)\bar{C}\eta - \bar{M}^{-1}(q)\bar{R}(\dot{q}) \quad (11)$$

Where, M is the inertial force being experienced by the body of the robot, B is the input transformation matrix, η is the state matrix for any instance and τ is the torque on the system.

3.2.2. Overall state space model

Although the dynamic and drive model are realized, the entire system calculation have to be done by involving both the dynamic and drive equations. This idea was well explained in the paper [1], while trying to device an optimum state space for simulation as shown in (12).

$$\begin{bmatrix} \dot{v}_x \\ \dot{\omega} \\ i_{a1} \\ i_{a2} \end{bmatrix} = A_1 \begin{bmatrix} v_x \\ \omega \\ i_{a1} \\ i_{a2} \end{bmatrix} + B_1 \begin{bmatrix} V_1 \\ V_2 \end{bmatrix} + D \quad (12)$$

Here, A_1 is a 4x4 matrix representing the Wheel Load Constants, B_1 is a 4x2 matrix representing Voltage Conversion Constant and D is a disturbance, which is of the order 4x1. The constants are in the form of a state space, which keep changing with respect to x_{ICR} as shown in (13-15).

$$A_1 = \begin{bmatrix} 0 & -x_{ICR}W & \frac{nK_i}{mr} & \frac{nK_i}{mr} \\ \frac{mx_{ICR}W}{mx_{ICR}^2+I} & 0 & \frac{-nK_i}{r(mx_{ICR}^2+I)} & \frac{nK_i}{r(mx_{ICR}^2+I)} \\ \frac{-nK_e}{rL_a} & \frac{nK_e}{rL_a} & \frac{-R_a}{L_a} & 0 \\ \frac{-nK_e}{rL_a} & \frac{-nK_e}{rL_a} & 0 & \frac{-R_a}{L_a} \end{bmatrix} \quad (13)$$

$$B_1 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \frac{1}{L_a} & 0 \\ 0 & \frac{1}{L_a} \end{bmatrix} \quad (14)$$

$$D = \begin{bmatrix} \frac{-F_{rx}}{m} \\ -x_{ICR}F_{ry} - M_r \\ \frac{mx_{ICR}^2+I}{0} \\ 0 \end{bmatrix} \quad (15)$$

These 4xN matrices are reduced to 2xN matrix by assuming L_a to be zero since it does not make any changes for the arguments as shown in (16-20).

$$u_{va} = R_a i_a + nK_e w_i \quad (16)$$

$$\begin{pmatrix} \dot{v}_x \\ \dot{\omega} \end{pmatrix} = A \begin{pmatrix} v_x \\ w \end{pmatrix} + B \begin{pmatrix} V_1 \\ V_2 \end{pmatrix} + D \quad (17)$$

$$A = \begin{bmatrix} \frac{2n^2 K_i K_e}{mr^2 R_a} & -x_{ICR}W \\ \frac{mx_{ICR}W}{mx_{ICR}^2+I} & \frac{-2n^2 c^2 K_i K_e}{r^2 R_a (mx_{ICR}^2+I)} \end{bmatrix} \quad (18)$$

$$B = \begin{bmatrix} \frac{nK_i}{mrR_a} & \frac{nK_i}{mrR_a} \\ -ncK_i & ncK_i \\ rR_a(mx_{ICR}^2+1) & rR_a(mx_{ICR}^2+1) \end{bmatrix} \tag{19}$$

$$D = \begin{bmatrix} \frac{-F_{rx}}{m} \\ -x_{ICR}F_{ry}-M_r \\ mx_{ICR}^2+1 \end{bmatrix} \tag{20}$$

General form of the state space representation is as shown in (21):

$$\dot{X} = A(X) X + BU + D(X) \tag{21}$$

Where the system states are represented by (22):

$$X = [v_x \ w]^T \tag{22}$$

The control input vector U is represented by (23):

$$U = [V1 \ V2]^T \tag{23}$$

D(X) is considered as a disturbance.

3.2.3. Design of LQR controller

The LQR controller can be designed for a system in two different ways. Homogenous LQR Controller: This type of LQR control does not take disturbance in factor. This is helpful as it gives faster response when the system is designed to move even terrain. Non Homogenous LQR controller: This LQR controller considers disturbances and it is a approach that is more practical. It helps the SSMR or a system move in uneven terrains. In Figure 11 is design of controller.

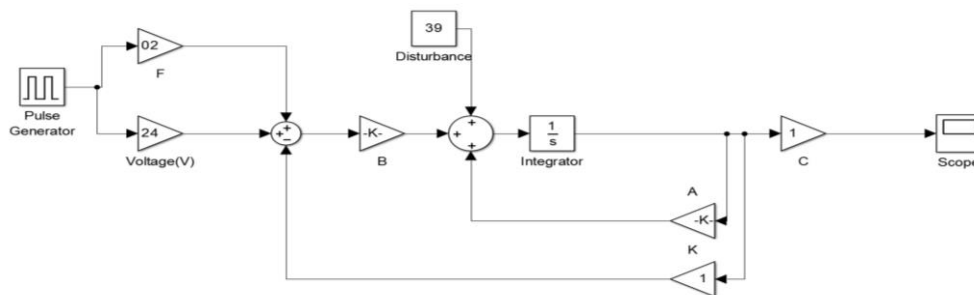


Figure 11. Design of controller

3.2.4. Homogenous LQR design

Consider the state space representation of a system (24-25):

$$\dot{x} = Ax + Bu, \tag{24}$$

$$y = Cx \tag{25}$$

With $x(t) \in R^n$,

$U(t) \in R^m$ and the initial condition is $x(0)$.

Assuming that all the states are measurable and seek to find a state-variable feedback control law as shown in (26):

$$u = -Kx \tag{26}$$

Here, K is the feedback gain vector. The quadratic cost function defined by (27):

$$J = \left(\frac{1}{2}\right) \int_0^{\infty} (x^T Q x + u^T R u) dt \quad (27)$$

The optimal feedback gain vector can be calculated by (28):

$$K = R^{-1} B^T P \quad (28)$$

Where P is the solution of the Algebraic Riccati Equation (ARE) defined by (29):

$$A^T P + P A + Q - P B R^{-1} B^T P = 0 \quad (29)$$

3.2.5. Non homogenous LQR design

The closed loop system equation of the controller is $\dot{x} = Ax + Bu$. $\dot{x} = (A - BK)x$. The function is given by (30-34):

$$V(x) = x^T P x \quad (30)$$

$$\dot{V} = \dot{x}^T P x + x^T P \dot{x} \quad (31)$$

$$\dot{V} = [(A - BK)x]^T P x + x^T P [(A - BK)x] \quad (32)$$

$$\dot{V} = x^T [(A^T P + P A + Q - P B R^{-1} B^T P) - Q - P B R^{-1} B^T P] x \quad (33)$$

$$\dot{V} = x^T [-Q - P B R^{-1} B^T P] x \quad (34)$$

From \dot{V} it is clear that $R > 0$, $P > 0$ and hence $P B R^{-1} B^T P > 0$. Also $Q > 0$. Hence $(Q + P B R^{-1} B^T P) > 0$. Therefore $\dot{V} < 0$. Hence the closed loop system is stable. Control Solution (35-37):

$$u = -R^{-1} B^T \lambda \quad (35)$$

$$u = -R^{-1} B^T (P x + K) \quad (36)$$

$$u = -R^{-1} B^T P x - R^{-1} B^T K \quad (37)$$

It is observed from the equation that even after the present state x tends to zero there is a residual controller acting which tends to reduce the disturbance.

3.2.6. Implementation of LQR on NXP LPC 1768 microcontroller

LPC 1768 controller has a 32-bit ARM Cortex-M3 core running at 96MHz. It also has 512KB flash, 32KB RAM and lots of I/O pins, which are required for the operation of SSMR.

Algorithm

Step 1 Initialize the pins of the microcontroller.

Step 2 Define parameters like gains, A , B , C , D , K

Step 3 Input the desired speed

Step 4 Calculate the count per second "t" for the desired speed

Step 5 Assign interrupts for the counts

Step 6 Receive the feedback from the encoder for time "t"

Step 7 Compare the desired speed and feedback for time "t"

Step 8 Calculate the gain matrices A , B , C , D , K , Q , R

Step 9 Calculate the control unit $u = (f+v) - Kx$

Step 10 Generate PWM using respective gains and errors as base PWM

Step 11 Run the entire step in loop.

***Note- D is taken as zero in case of homogenous LQR controller.**

4. RESULTS

Design Results of the Skid Steer Mobile Robot. Table 1 shows the final Electrical and Mechanical parameters of the SSMR robot. Detail Calculation of SSMR is given in APPENDIX-1. Table 2 shows the final results of the parameters of SSMR.

Table 1. Electrical and Mechanical Parameters of the Robot

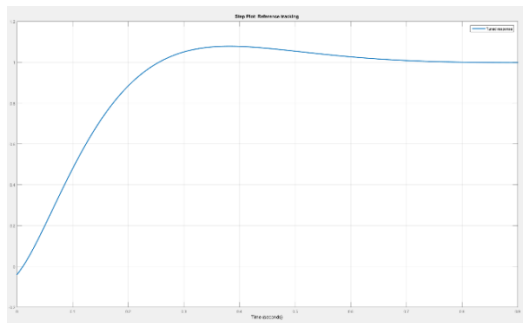
Parameters	Values (Dimension)
R (radius of wheel)	0.1075 (m)
M (Mass of Robot)	46 (kg)
I (Moment of Inertia about COM)	0.006288 (Kg m ²)
R _a (Armature Resistance)	0.317 (Ohms)
K _t (Torque Constant)	0.03 (Nm/A)
K _e (EMF Constant)	0.03 (Vs/rad)
a (Lateral distance of rear wheel center from COM)	0.2275 (m)
b (Lateral distance of front wheel center from COM)	0.2275(m)
c (Lateral distance between COM and the side wheel)	0.2275(m)
x _{ICR} (X coordinate of Instantaneous Centre of Rotation)	-0.15

Table 2. Final calculated Parameters of SSMR

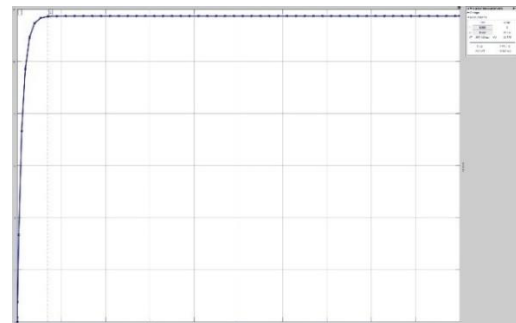
Parameters	Value
A	1.248
B	10.83
C	1
D	0
F	6.173 N
K	1.54
V	24 V

4.1. Performance comparison result

In Figure 11 is (a), (b) and (c) shows the response of PID and LQR-Homogeneous and non-Homogeneous Controller.



(a) Response of PID Control



(b) Response of Homogenous LQR Controller



(c) Response of Non Homogenous LQR controller

Figure 11. (a), (b) and (c) shows the response of PID and LQR-Homogeneous and non-Homogeneous Controller

From the results in Table 3, it is observed that the LQR with Homogeneous Controller gives the better performance in settling time, peak amplitude and overshoot.

Table 3. Performance Comparison

Time Specifications	PID	LQR(H)	LQR (NH)
Settling Time	1.09	0.596	0.692
Peak Amplitude	1.95	0.71	0.996
Overshoot	13.1%	0.1366	6.1366%

5. HARDWARE IMPLEMENTATION OF LQR ON SSMR USING NXP LPC 1768

Both PID and LQR Controllers are implemented on the NXP LPC 1768 for the Skid Steering Mobile Robot and the performance comparison is made. The result in Table 4 demonstrates the following:

- The LQR Controller of the SSMR Robot has larger velocity than the traditional PID controller.
- The reference velocity of the system was set at 20 m/s, The LQR control gave practically very less loss in comparison to the PID Controller.
- Further observation shows that the velocity of all the wheels are the same for LQR Controller.

Hence based on our theoretical modeling supported by Experimental results, it can be concluded that LQR Controller is an Optimal Controller. Performance Comparison with Embedded Board NXP LPC 1768 as shown in Table 4.

Table 4. Performance Comparison with Embedded Board NXP LPC 1768

Q4 Encoders	CONTROLLERS	WHEELS			
		LEFT FRONT	LEFT REAR	RIGHT FRONT	RIGHT REAR
VELOCITY	PID	13	13.3	14.6	14.7
	LQR	15.9	15.9	16	16
COUNTS	PID	29000	29000	29000	29000
	LQR	29000	29000	29000	29000

ACKNOWLEDGEMENT

Our sincere thanks goes to the Vision Group of Science and Technology (VGST), Karnataka to acknowledge our research and provide us the financial support to carry out the research at NMIT. We express our sincere thanks to our colleagues at the Robotics Research Center, NMIT for providing support. Finally, our sincere gratitude goes to Prof. N R Shetty, Director NMIT and Dr. H C Nagaraj, Principal NMIT for providing the infrastructure support and wholehearted encouragement to carry out the research at NMIT.

REFERENCES

- [1] Osama Elshazly, Ahmed Abo-Ismael, Hossam S. Abbas and Zakarya Zyada, "Skid-steering mobile robot Modeling and Control," 2014 UKACC., 978-1-4799-5011-9/14 ©2014.
- [2] K. Kozłowski and D. Pazderski, "Modeling and control of a 4-wheel Skid-steering mobile robot," *Int. J. Appl. Math. Comput. Sci.*, vol. 14, no. 4, pp. 477–496, 2004.
- [3] A. Mandow, J. L. Martinez, J. Morales, J. L. Blanco, A. G. Cerezo, and J. Gonzalez, "Experimental kinematics for wheeled skid-steer mobile robots," in *In Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems, USA, 2007*, pp. 1222–1227.
- [4] L. Caracciolo, A. D. Luca, and S. Iannitti, "Trajectory tracking control of a four-wheel differentially driven mobile robot," in *In Proc. of IEEE International Conference on Robotics and Automation, USA, 1999*, pp. 2632–2638.
- [5] K. Ogata, *Modern Control Engineering*, 5th ed. Pearson Prentice-Hall, 2010.
- [6] O. D. I. Nwokah and Y. Hurmuzlu, *The Mechanical Systems Design Handbook*, CRC Press, 2002, ch 26.

APPENDIX

Detail calculation of SSMR Robot is given below:

$$A = \begin{pmatrix} 0.125998 & 0.015 \\ -0.6576 & 1.9064 \end{pmatrix}$$

$$A = 1.2486$$

$$B = \begin{pmatrix} 0.7369 & 0.7369 \\ -7.3499 & 7.3499 \end{pmatrix}$$

$$B = 10.83$$

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

$$C = 1$$

$$D = \begin{pmatrix} -0.3924 & \\ -5.45 & \end{pmatrix}$$

$$D = 39$$

The gain matrix are obtained using Ricatti equation and matrices obtained from SSMR parameters i.e. The value of Q matrix is

$$Q = \begin{pmatrix} 0.1 & 0 \\ 0 & 0 \end{pmatrix}$$

Therefore after solving the Ricatti Equation, the Gain Matrix is:

$$K = \begin{pmatrix} 1.2187 & -0.6093 \\ 1.2187 & 0.6093 \end{pmatrix}$$

DC Motor

Maxon DC motors are high-quality motors fitted with powerful permanent magnets. The “heart” of the motor is the worldwide-patented ironless rotor. This type of motor provides motor-gear head combinations for applications in extreme environments in terms of temperature, pressure also. The types of gear heads available are: Spur gear heads, planetary gear heads made of plastic, ceramic (zirconium dioxide and aluminum) or stainless steel. The motor used is having 12V power supply. The maxon motor as shown in *Figure 1*.



Figure 1. Maxon Motor

Hercules Motor Driver

Hercules 6V-36V, 15Amp Motor Driver can take up to 30A peak current load and can be operated up to 10 KHz PWM. Motor driver can be interfaced with 3.3V and 5V logic levels. Motor driver has built-in protection from under/over voltage, over temperature and short. Motor Driver has optional ACS714 current sensor for current sensing. You can choose current sensor installation option at the time of placing the order. The Motor driver has terminal block as power connector and 10 pin 2510 type relimate connector for

the logic connection. It is suitable for high performance robots, Robocon, Robo-cup, US First, Battle robots levels. The Hercules motor driver as shown in *Figure 2*.

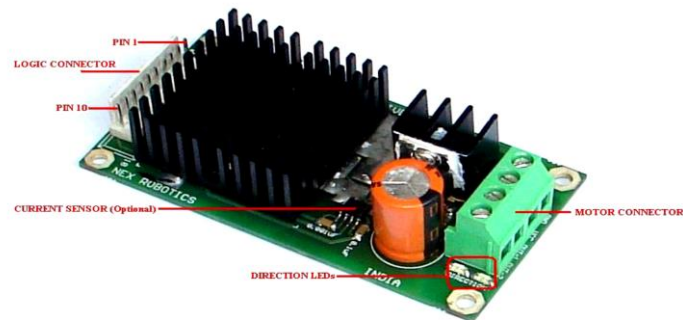


Figure 2. Hercules Motor Driver

BIOGRAPHIES OF AUTHORS



Dr. Jharna Majumdar is currently the Dean R & D, Prof. & Head Dept of M Tech CSE and Head, Center for Robotics Research at the NitteMeenakshi Institute of Technology, Bangalore. Dr. Majumdar served DRDO, Min. of Defense, Govt. of India from 1990 to 2007 and retired as Scientist G and Head of Aerial Image Exploitation Division, Aeronautical Development Establishment, Bangalore, India.



Mr. Sudip Chandra Gupta is Final Year Electrical & Electronics Engineering Student of NitteMeenakshi Institute of Technology and working at the Center for Robotics Research under the guidance of Dr. Jharna Majumdar. Their main area of research interest is Linear and Non Linear Controller for Autonomous Mobile Platform.



Mr. B Prassanna Prasath is Final Year Electrical & Electronics Engineering Student of NitteMeenakshi Institute of Technology and working at the Center for Robotics Research under the guidance of Dr. Jharna Majumdar. Their main area of research interest is Linear and Non Linear Controller for Autonomous Mobile Platform.