A New Approach for Control of Two-wheeled Mobile Robot

Zahra Khajenasir Jahromi, Hamed Kharrati Shishavan*

Faculty of Electrical & Computer Engineering, University of Tabriz, Iran

Copyright©2019 by authors, all rights reserved. Authors agree that this article remains permanently open access under the terms of the Creative Commons Attribution License 4.0 International License

Abstract In this paper, a study of the GBOT1001 two wheeled mobile robot is presented. Linearized dynamic equations are used to design a controller for the inherently unstable mobile robot. A Fuzzy controller is designed for the robot to direct the orientation of the chassis. Then linear quadratic regulator is also incorporated into the closed-loop system. The parameters of the LQR controller are computed first with the trial and error methods then, with the aid of the Genetic Algorithm. The resulting controller is referred to as the *Fuzzy-TE-LQR* and *Fuzzy-GA-LQR* controller respectively. The closed-loop responses for both controllers are obtained via computer simulations. Simulation results show that the *Fuzzy-GA-LQR* controller exhibits a less oscillatory closed-loop response that is achieved with a reasonable control effort.

Keywords Two-wheeled Self-Balancing Robot, Fuzzy Controller, Linear Quadratic Regulator, Genetic Algorithm, Stability

1. Introduction

Two-wheeled mobile robots are characterized by two driving wheels connected to an intermediate body carrying actuation, transmission, sensor, control, and communication subsystems [1]. With a center of mass above the wheel axles, these robots must actively stabilize themselves to prevent toppling. Their wheel configuration makes them highly maneuverable due to their ability to turn on the spot; however, they are more difficult to control than statically stable wheeled robots [2]. While the inherent unstable dynamics of the system has made two-wheeled mobile robots an appealing subject for laboratory research, the high maneuverability has justified their use in industry where easy navigation in various terrains is of interest. Numerous control methods have been introduced to deal with the instability of two-wheeled mobile robots.

Combination of a robust pole-placement design and a linear quadratic regulator [3], a linear quadratic regulator optimized by a particle swarm algorithm [4], a Fuzzy Immune PD controller and robust sliding mode controllers for accurate tracking and disturbance rejection [5, 6]. Are some examples.

In this paper, a study of the GBOT1001 two wheeled mobile robot manufactured by Googol Technology Ltd¹, is presented. The control objective of the Two-wheeled mobile robots is to perform motion and speed control of the wheels while stabilizing the pendulum around the upright position (unstable equilibrium point). This type of systems that have less numbers of actuators than the degrees of freedom to be controlled, are called “under-actuated systems” [5]. In studies carried out by Wu, Zhang and Wang on this robot [7,8,9], a linear model for the robot is derived and a variety of controllers such as Pole Placement and the LQR controller [7], fuzzy controller [8] and Fuzzy-PD controller [9] are designed for the linearized model. The present paper intends to introduce a modified fuzzy controller that serves as a variation on the previously introduced fuzzy controller [8,9] for the GBOT1001 two-wheeled mobile robot. The proposed fuzzy controller is then combined with a linear quadratic regulator, the parameters of which are tuned with the aid of a Genetic Algorithm. The designed fuzzy controller is also combined with the LQR of previous papers [7,8], so that the effectiveness of the tuning can be observed in computer simulations.

In section IIA of the present paper, a brief description of the robot is provided and the derivation of a linearized mathematical model for the system is revisited in section IIB. This model is controlled with an LQR controller (TE and GA). In section IIIA, with establishing the model of system, suing fuzzy logic control theory to control the orientation of the robot, is introduced. A linear quadratic

¹ The company is located in the entrepreneur center of Hong Kong University of Science and Technology.

regulator that is based on a trial and error approach [7,8,9]. is reviewed in section IIIB. A new LQR controller based on Genetic Algorithm is also proposed. Genetic Algorithm mimics the process of evaluation and can be applied to optimize the process of controllers using natural operators. The concept and design procedure of Genetic Algorithm as an optimization tool is discussed in section IIIC [10]. Additionally Fuzzy control is implemented. The designed fuzzy controller is combined with both LQR controllers in sections IIB and IIC, resulting in a *Fuzzy-TE-LQR* controller and a *Fuzzy-GA-LQR* controller respectively. Simulation results for both controllers are presented in section IV and the effectiveness of the later controller is observed. The paper finally concludes in section V.

2. System Description and Mathematical Modeling

2.1. System Description

Figure 1. GBOT1001 of Googol Technology Ltd

The GBOT1001 two wheeled mobile robot studied in this paper (Fig. 1) is composed of a chassis, a vertical bar, a DC motor for each wheel, planetary gearboxes coupled to the wheels, a DSP board² implementing the controller, power amplifiers for the DC motors, odometry sensors and a receiver for the radio control unit [7,8,9].

2.2. Mathematical Model of the System

In order to derive a linear model for the robot, it is assumed that the robot is a rigid body with completely analogous wheels. It is also assumed that the cornering forces, friction and the motor dynamics are negligible. The two-wheeled mobile robot is therefore reduced to a mobile

inverted pendulum system. Figure 2 shows the free body diagram of the robot.

The linear movement of the chassis is characterized by the *position* x and the *linear speed* $\dot{x}$. Rotations about Z and Y axes are described by θ and δ , respectively, and the corresponding angular velocities are denoted by $\dot{\theta}$ and $\dot{\delta}$. In addition, m is the mass of each wheel and M is the mass of the chassis. Moreover, R is the radius of the wheels; L is the distance between Z axis and the center of gravity of the chassis and D is the distance between the centers of the wheels. For convenience, θ will be referred to as *orientation* throughout the paper.

The nonlinear dynamic equations governing the wheels and the chassis are linearized about the operating point $\theta = 0$. The resulting sets of linear state space equations are derived, after some calculations, as follows [7,8,9].

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{\theta} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & a_{23} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & a_{43} & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \theta \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ b_2 \\ 0 \\ b_4 \end{bmatrix} [C_\theta] \quad (1)$$

$$\begin{bmatrix} \dot{\delta} \\ \ddot{\delta} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \delta \\ \dot{\delta} \end{bmatrix} + \begin{bmatrix} 0 \\ b_6 \end{bmatrix} [C_\delta] \quad (2)$$

where

Figure 2. Free Body Diagram of the Robot

$$a_{23} = g \left(1 - \frac{4}{3} L \frac{M}{K_1} \right),$$

$$a_{43} = \frac{gM}{K_1},$$

$$b_2 = \left(\frac{4LK_2}{3K_1} - \frac{1}{ML} \right),$$

$$b_4 = -\frac{K_2}{K_1},$$

$$b_6 = \frac{6}{(9m + M)RD},$$

and

$$K_1 \stackrel{\text{def}}{=} \frac{1}{3} \frac{M(M+m)L}{M + \frac{3}{2}m}, \quad K_2 \stackrel{\text{def}}{=} \frac{M}{(M + \frac{3}{2}m)R} + \frac{1}{L}.$$

It can be seen from Eq. 1 and Eq. 2 that the state space

²The DSP board consists of a floating-point DSP from Analog Devices, an FPGA, three 10-bit D/A converters, and four 12-bit A/D converters.

equations for the two-wheeled mobile robot are written as two decoupled systems. Equation 1 describes the rotation of the two-wheeled mobile robot about the Z axis and Eq. 2 describes a rotation about the Y axis. C_θ is the sum of the left and right wheels' torques (*i.e.* $C_L + C_R$) and C_δ is their difference (*i.e.* $C_L - C_R$). It should be noted that the controllers designed in section III of the present paper, will make use of Eq. 1 and not Eq. 2. The following numerical values are provided for GBOT1001 by the manufacturer [7,8,9].

$$M = 21\text{kg}, m = 0.420\text{kg}, R = 0.106\text{m},$$

$$L = 0.3\text{m}, g = 9.8\text{m/s}^2$$

Incorporating the above values into Eq. 1 and Eq. 2, gives the final equations for the mathematical model of the system.

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{\theta} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & -26.25 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 90.125 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \theta \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ 2.0296 \\ 0 \\ -5.4708 \end{bmatrix} [C_\theta] \quad (3)$$

$$\begin{bmatrix} \dot{\delta} \\ \ddot{\delta} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \delta \\ \dot{\delta} \end{bmatrix} + \begin{bmatrix} 0 \\ 5.1917 \end{bmatrix} [C_\delta] \quad (4)$$

3. Controller Design

The controller design and stabilization are based on the accurate mathematical model without considering any uncertainties. However, uncertainties and model mismatch between the nominal model and the real plant is inevitable [5]. A variation on the fuzzy controller introduced in previous papers [8,9], is proposed. In section IIIA, unlike previous papers, in which *position* (x) and its derivative ($\dot{x}$)

were the sole inputs to the fuzzy controller, the present paper incorporates orientation (θ) into the fuzzy controller to restrain the robot from falling down successfully. Section IIIB reviews the design procedure of the LQR controller that had been designed for the two-wheeled mobile robot of Eq. [7,9]. In order to obtain optimal values for the weighing matrices of the linear quadratic regulator in section IIIB, a Genetic Algorithm approach is introduced in section IIIC.

3.1. Fuzzy Logic Controller

Fuzzy logic is the logic on which fuzzy control is defined. It is much similar in spirit to human thinking and natural language speaking than the traditional logical systems. Essentially, it provides an efficient means of capturing the approximate, inexact nature of the real world. Viewed in this perspective, the indispensable part of the fuzzy logic controller is a set of linguistic control rules linked by the dual concepts of fuzzy implication and the compositional rule of inference. In essence, fuzzy logic controller provides an algorithm which can transform the linguistic control strategy based on expert knowledge into an automatic control strategy [11].

The self-balancing robot control system which is based on the fuzzy control can overcome the instability and nonlinear nature of the system, but the fuzzy rules are too many, increasing difficulty of the reasoning. Therefore, the fuzzy control and LQR controller will be combined in this paper. It improves the performance of the controller greatly and it can be seen that this method reveals the best static and dynamic performance through the simulation tests and comparisons [12].

Figure 3. The Illustration of a Sample Rule for the Fuzzy Controller: Orientation Error and Position Error are Compensated in Stages I and II, Respectively.

Table 1. Designated Rules for the Proposed Fuzzy Controller

		POSITION								
		NL	NB	NM	NS	ZE	PS	PM	PB	PL
ORIENTATION	NL	NL	NL	NL	NL	NL	NL	NL	NL	NL
	NB	NB	NB	NB	NB	NB	NB	NB	NB	NB
	NM	NM	NM	NM	NM	NM	NM	NM	NM	NM
	NS	NS	NS	NS	NS	NS	NS	NS	NS	NS
	ZE	PS	PS	PS	PS	ZE	NS	NS	NS	NS
	PS	PS	PS	PS	PS	PS	PS	PS	PS	PS
	PM	PM	PM	PM	PM	PM	PM	PM	PM	PM
	PB	PB	PB	PB	PB	PB	PB	PB	PB	PB
	PL	PL	PL	PL	PL	PL	PL	PL	PL	PL

For the two-wheeled mobile robot at hand, a Mamdani Fuzzy Inference System with triangular membership functions and fuzzy sets NL (Negative Large), NB (Negative Big), NM (Negative Medium), NS (Negative Small), ZE (Zero), PS (Positive Small), PM (Positive Medium), PB (Positive Big) and PL (Positive Large) is designed. Position (x) and orientation (θ) are used to derive the fuzzy rules. In deriving the fuzzy rules, the orientation is dealt with first; that is, the position error is compensated only after the orientation error approaches zero (ZE). In order for the orientation error to approach zero, it is assumed that the torque applied to the wheels must act in the same direction that the chassis deflects. Figure 3 illustrates one of the assigned rules to the orientation and position. If the chassis (i.e. the inverted pendulum of Fig. 3) deviates from the desired ZE orientation toward, say NL, the wheels must move to the left thereby justifying the use of NL as the output value for the wheels' torque regardless of the position of the robot. Now with the orientation error having approached zero, the position error (i.e. NM in Fig. 3) is gently compensated. The resulting fuzzy rules are shown in Table 1.

3.2. Trial and Error LQR (TE-LQR)

In the optimal control problem for a linear system with the state-space equation $\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t)$, it is desired that the performance index given by $J = \int_0^\infty [\mathbf{x}^T(t)\mathbf{Q}\mathbf{x}(t) + \mathbf{u}^T(t)\mathbf{R}\mathbf{u}(t)] dt$ be minimized. The Linear Quadratic Regulator (LQR) that fulfills this criterion is $\mathbf{u}(t) = -\mathbf{K}\mathbf{x}(t)$. $\mathbf{K}$ is the state feedback gain matrix of this optimal control law and $\mathbf{Q}$ and $\mathbf{R}$ are weighing matrices. $\mathbf{Q}$ is a positive definite (or positive semi-definite) Hermitian or real symmetric matrix; $\mathbf{R}$ is a positive definite Hermitian or real symmetric matrix [13].

As for the two-wheeled mobile robot represented by Eq. 3, the proper state feedback gain matrix $\mathbf{K}$ is found by MATLAB® command `lqr(A,B,Q,R)` to be

$$\mathbf{K} = \begin{bmatrix} -31.6228 & -24.4086 & -84.1597 & -14.2010 \end{bmatrix}.$$

Furthermore, the weighing matrices $\mathbf{Q}$ and $\mathbf{R}$ have been chosen with trial and error to be [7,9].

$$\mathbf{Q} = \begin{bmatrix} 1000 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1000 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\mathbf{R} = 1.$$

Due to the trial and error nature of the selection of the weighing matrices $\mathbf{Q}$ and $\mathbf{R}$, the resulting regulator will be referred to as the *Trial and Error LQR* and is abbreviated to *TE-LQR*.

3.3. Genetic Algorithm Tuned LQR (GA-LQR)

Invented in the early 1970s, Genetic Algorithms have gained considerable popularity as general-purpose robust optimization and search techniques. Genetic Algorithms are particularly attractive because instead of a naïve “search and select” mechanism, they use crossover to exchange information among existing solutions to locate better solutions [14]. The feasibility of Genetic Algorithms (GAs) is based on Darwinian's theory of survival of the fittest. Genetic Algorithms start with a set of solutions represented by chromosomes, called population. Solutions from one population are taken and used to form a new population, which is motivated by the possibility that the new population will be better than the old one. Furthermore, solutions are selected based on their fitness to form new solutions. This process is repeated until a certain condition is satisfied. Genetic Algorithms can be applied when there is insufficient knowledge of the system or high complexity is there. It can find optimal solutions in the search space with special operators like crossover and mutation [10].

In the case of the LQR controller in section IIIA, the performance index of $J = \int_0^\infty [\mathbf{x}^T(t)\mathbf{Q}\mathbf{x}(t) + \mathbf{u}^T(t)\mathbf{R}\mathbf{u}(t)] dt$ needs to be minimized in the optimization process of the Genetic Algorithm and the corresponding $\mathbf{Q}$ and $\mathbf{R}$ matrices are going to be obtained. Global Optimization Toolbox of MATLAB® is utilized for this purpose; the command ‘gaoptimset’ customizes the parameters of the Genetic Algorithm and the command ‘ga’ executes the algorithm. $\mathbf{Q}$ is assumed to be a diagonal matrix and $\mathbf{R}$ a scalar. The

arrays of the Q matrix and the scalar R are taken to be greater than 0.001. The number of 'generations' is assumed to be 10. Some other parameters also need to be adjusted: The function that creates the initial population for the Genetic Algorithm (i.e. 'Creation function') is chosen to be a Feasible one (i.e. @gacreationlinearfeasible). The mutation option that specifies how the Genetic Algorithm makes small random changes to the individuals in the population and enables the Genetic Algorithm to search a broader space is chosen to be Adaptive Feasible (i.e. @mutationadaptfeasible). The crossover option that specifies how the Genetic Algorithm combines two individuals or parents to form a crossover child for the next generation is chosen to be the weighted arithmetic mean of two parents (i.e. @crossoverarithmetic). The @fmincon option of the Hybrid function group is also utilized to perform constrained minimization after the Genetic Algorithm terminates. The remaining parameters for the Genetic Algorithm are the default values implemented by the Toolbox and are pointed out in MATLAB® Product Help as well as the product's website³. Applying the algorithm, yields a minimum of $J = 2.2729e-5$ for the performance index thus resulting in the following weighing matrices,

$$Q = \begin{bmatrix} 5.3510 & 0 & 0 & 0 \\ 0 & 3.1106 & 0 & 0 \\ 0 & 0 & 0.1274 & 0 \\ 0 & 0 & 0 & 2.4158 \end{bmatrix}$$

$$R = 0.001.$$

The state feedback gain matrix K is therefore found to be

$$K = [-73.1518 \quad -99.7916 \quad -352.7827 \quad -91.4542].$$

The resulting regulator will be referred to as the *Genetic Algorithm Tuned LQR* and is abbreviated to *GA-LQR* in the remainder of the paper.

4. Simulations

In this section, the simulation results for the *Fuzzy TE-LQR* and *Fuzzy GA-LQR* controllers will be presented. Figure 4 shows the Simulink® block diagram of the system. As it can be seen from the figure, Eq. 3 governing the dynamics of the two-wheeled mobile robot is embedded in a subsystem with the wheels' torque as its input and the state vector of $[\mathbf{x} \quad \dot{\mathbf{x}} \quad \boldsymbol{\theta} \quad \dot{\boldsymbol{\theta}}]^T$ as its output. In addition, position and orientation are fed as the inputs to the fuzzy controller and the linear quadratic regulator is incorporated to the closed-loop system as a state feedback torque controller. A square pulse disturbance torque with amplitude of 0.1 N.m. and duration of 0.1 second is applied

and the response of the closed loop system is observed.

Figure 4. The Closed-loop System with the Proposed Controller

Simulation results for position, orientation, linear speed and angular velocity for both *Fuzzy TE-LQR* and *Fuzzy GA-LQR* controllers with zero initial conditions are shown in Fig. 5. It can be seen from the figure that both controllers exhibit robustness in the presence of the disturbance torque. However, the *Fuzzy TE-LQR* controller proves to be a more oscillatory response as expected. Figure 6 shows the control signal for both controllers. As it can be seen, the *Fuzzy GA-LQR* proves to be more energy efficient compared to the *Fuzzy TE-LQR* controller.

Figure 5. Simulation Results for Fuzzy GA-LQR and Fuzzy TE-LQR Controllers

Figure 6. Simulation Results for Control Signals of Fuzzy GA-LQR and Fuzzy TE-LQR Controllers.

The proposed controllers are also evaluated with nonzero initial conditions for position and orientation. An initial value of 0.01 m for position and 0.00 008 rad. for orientation are considered. The resulting simulation results for position, orientation, linear speed and angular velocity as well as the control signals are depicted in Fig.7 and Fig.8.

As it can be seen from Fig. 7, the robustness of both controllers is reconfirmed with *Fuzzy GA-LQR* controller's response being less oscillatory. Similarly, in terms of energy consumption, the *Fuzzy GA-LQR* controller of Fig. 8 once again proves to be more efficient.

Figure 7. Simulation Results for Fuzzy GA-LQR and Fuzzy TE-LQR Controllers with Nonzero Initial Conditions for x and θ

Figure 8. Simulation Results for Control Signals of Fuzzy GA-LQR and Fuzzy TE-LQR Controllers with Nonzero Initial Conditions for x and θ

5. Conclusions

The simulation results present that the self-balancing robot system has the characteristics of fast response speed, fewer oscillation times and small overshoot in the best performance. The *Fuzzy-GA-LQR* controller designed in this paper for the GBOT1001 mobile robot showed these capabilities. In designing the Fuzzy controller, the priority was given to the adjustment of the orientation and exhibited robustness in the presence of the disturbance.

Advantages of optimal control are obvious from simulation results using Linear Quadratic Regulator method. Genetic Algorithms are effective techniques of swiftly exploring a reasonable solution for an intricate problem. The oscillations in the closed-loop response fade in the presence of the Genetic Algorithm. Combination of Fuzzy Controller, Genetic Algorithms and Linear Quadratic Regulator is a highly reliable controller to unstable nonlinear systems and guarantees closed loop stability and robustness. The controller is confined to gentle alterations

in position and orientation of GBOT1001 mobile robot. The utilization of nonlinear dynamic equations in the design procedure of the controller can therefore be a subject for future research.

REFERENCES

-
- [1] Salerno, A. & Angeles, J. Modeling and Controllability of Two-Wheeled Quasi-Holonomic Robots, Montreal, Canada: Centre for Intelligent Machines, 2005.
 - [2] Ping Man Chan, R., A. Stol, K. & Roger Halkyard, C. Review of modelling and control of two-wheeled robots. Annual Reviews in Control, 2013, Volume 37, pp. 89-103.
 - [3] Amano, Y. Stability Control for Two-Wheeled Mobile Robot Using Robust Pole-Placement Method. Proceeding of the 2nd International Conference on Intelligent Technologies and Engineering System, Springer International Publishing Switzerland, 2014, Volume 293, pp. 259-268.
 - [4] Fang, J. The Research on the Application of Fuzzy Immune PD Algorithm in the Two-Wheeled and Self-Balancing Robot System. International Journal of Control and Automation, 2014, 7(10), pp. 109-118.
 - [5] Jian-Xin Xu, Zhao-Qin Guo, Tong Heng Lee. Design and Implementation of Integral Sliding-Mode Control on an Underactuated Two-Wheeled Mobile Robot. IEEE Transactions on Industrial Electronics, 2014, 61(7).
 - [6] A. Sinha, P. Prasoon, P. K. Bharadwaj, A. C. Ranasinghe. Nonlinear Autonomous Control of a Two-Wheeled Inverted Pendulum Mobile Robot Based on Sliding Mode. International Conference on Computational Intelligence & Networks, 2015.
 - [7] Wu, J. & Zhang, W., March. Research on Control Method of Two-wheeled Self-balancing Robot. Shenzhen, China, s.n, 2011.
 - [8] Wu, J. & Zhang, W., August. Design of Fuzzy Logic Controller for Two-wheeled Self-balancing Robot. Harbin, China, s.n, 2011.
 - [9] Wu, J., Zhang, W. & Wang, S. A Two-Wheeled Self-Balancing Robot with the Fuzzy PD Control Method. Mathematical Problems in Engineering, 2012, Volume12, pp. 1-13.
 - [10] Malhotra, R., Singh, N. & Singh, Y. Genetic Algorithms Concepts, Design for Optimization of Process Controllers. Computer and Information Science, 2011, 4(2), pp. 39-54.
 - [11] Lee, C. C. Fuzzy Logic in Control Systems: Fuzzy Logic Controller - Part I. IEEE Transactions on Systems, Man, and Cybernetics, 1990, 20(2), pp. 404-418.
 - [12] Fang, J. The LQR Controller Design of Two-Wheeled Self-Balancing Robot Based on the Particle Swarm Optimization Algorithm. Mathematical Problems in Engineering, 2014.
 - [13] Ogata, K. Modern Control Engineering. s.l.:Prentice Hall, 2002.
 - [14] Srinivas, M. & Patnaik, L. M. Genetic Algorithms: A Survey. Computer Journal of IEEE Computer Society, 1994, 27(6), pp. 17-26.