

STABILITY CONTROL MODELLING UNDER DYNAMIC MOTION SCENARIO OF A DIFFERENTIAL DRIVE ROBOT

ADEDOTUN O. OWOJORI^{1*}, KAYODE F. AKINGBADE¹, WALIU O. APENA¹,
ERASTUS O. OGUNTI¹

¹*Department of Electrical and Electronics Engineering, Federal University of Technology
PMB 704, Akure, Ondo State, Nigeria*

Abstract: Intelligence incorporated in many devices makes it easier to achieve self-balancing and autonomous driving in differential drive robot. Basically, differential drive robotic system describes an unstable, nonlinear system related to an inverted pendulum. The research attempts to harness the parameters obtained from a computer-aided design tool (Solid works) to model the system for complete stability control and dynamic motion of the system within a planned trajectory. A linearized dynamic equation is obtained for the overall system design of a mobile robot, and the linear quadratic regulator concept is adopted to obtain an optimum state feedback gain. The simulation results are obtained on MATLAB software interfaced with an Arduino board with deployable sensor technologies. Scenarios of disturbance would be simulated to ascertain the stability conditions of the system at static position or dynamic position. Signal analysis and computer vision techniques serve as leverage to make the design achievable. Localization and navigation referred to as tracking a planned trajectory or moving through paths filled with obstacles in a given space are also included.

Keywords: differential drive robots, autonomous, self-balancing, CAD; MATLAB/Simulink

1. INTRODUCTION

The concept of robotics initially hinges on the classical technique of bringing functional parts together to achieve a unified task which could be mechanical based, that is following the conservation of energy or electrical-based, that is adopting actuators among others [1-2]. These robots in a sense are concerned with synthesizing human functions and based on most classical attempt, robot hobbyist tries to model a generalized mathematical equation to depict the behaviour of the robot and further extend it to carry out other tactical functions.

Though this method had been exploited with little success in holistically developing a mathematical model of most complex systems hence, the need for a more suitable method where the system can learn given few training data or boundary value data [3]. Active researches in this area discuss the use of Artificial Intelligence (AI) introduced into most systems which in turn requires ample data set to effectively interpret the behaviour of the system [3]. A more hybridized approach of the two methods could give significant improvement in designing most complex robotic systems. This involves a well-developed mathematical model of the robot in conjunction with a Computer Aided Design (CAD) tool to estimate allowable parameter tolerance and application software (MATLAB/Simulink) which matches the experimental results with the developed model [4].

Robotic systems are featured in consumer robots (artificial intelligence (AI)-Powered robot assistance), lightweight robots, industrial robots, complex space project robots among others, which are capable of giving more

* Corresponding author, email: aowojori@futa.edu.ng

regulation, satisfactory sensitivity, stability and efficiency. [5] describes a new frontier of collaborative robots which allows synch between human teleoperation and complementing robots.

The field of robotics as an area of research is increasingly gaining prominence as the technologies applied in their design grows from the basic theoretical stability concepts of robotic dynamics to more advanced ones targeted towards self-coordinated or autonomous systems, path profile navigation algorithms and feedback control in differential drive or balancing robots. [6] proposed a least square method for non-analytical objective function using redundancy to improve the performance of motion, [7] describes the trajectory planning and tracking control of a 6 degree of freedom rigid robot using a general constrained nonlinear optimization, [8] proposed the use of Linear Quadratic Regulator (LQR) for state feedback and Kalman filter for state estimation of a self-balancing two-wheeled robot design.

The wheeled robot, in particular, has been shown great interest by hobbyist and researchers as it studies how well a nonlinear and unstable system can be controlled to achieve cutting edge tasks such as increased degree of freedom, trajectory planning, tracking and self-balancing tasks [2, 8-11], and use in navigating highly volatile environment [12]. Wheeled robots are inspired by biological systems whose attribute makes it possible to develop robotic systems which could move through its environment autonomously deciding how, when and where to move [6]. These operations are achieved based on physical interfacing of wheel and other actuators and sensors with a controller. It turns out that the robot rolls over around the wheel axis when no external control is introduced to it and tries to maintain a vertical position with tilt angle 0^0 in the presence of a controller.

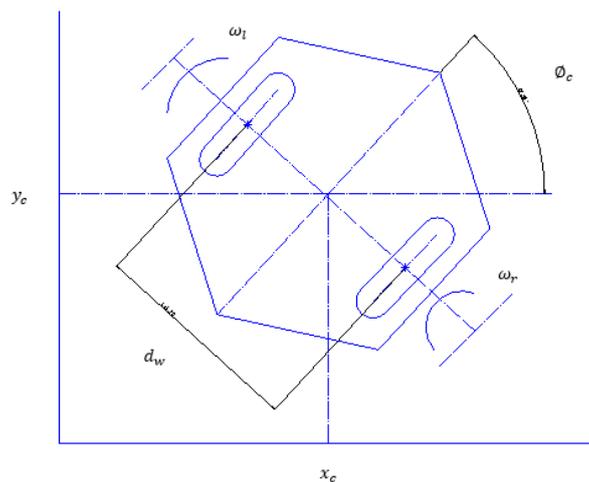


Fig. 1. Differential drive robot.

The mobile robot shown in Figure 1 was first modelled mathematically, having in mind the two variations of the differential drive system that exist, that is; a system with the wheel axis aligning with the center of gravity and the other, having a distance 'd' away from the center of gravity but augmented with a castor wheel or steering wheel. The model is supported by visual representation on SolidWorks (CAD software) from which allowable design tolerances of the system are generated and exported to MATLAB/ Simulink software. The simulation and physical implementation of the system are then carried out to obtain simulated and experimental results which are matched with the mathematical and visualization model. This is followed by control analysis of the system on MATLAB/ Simulink before finally deployed on a hardware system of choice (Arduino Microcontroller) to establish its appropriate real-time system behavior.

2. EXPERIMENTAL SETUP

The conceptualized two-wheel physical model is designed with a wooden base of dimension 15 cm x 14 cm x 1.5 cm, weight 0.15 kg, wheel axis aligning with the center of gravity of the base and the wheels placed at midpoints of the 15 cm length, and augmented with a NEMA17 stepper motor and wooden wheels shown in Figure 2.



Fig. 2. Developed differential drive robot without controls.

2.1. System modelling

The model includes the mathematical representation of static and dynamic equations. The kinematic equations of differential drive robots are given in equations (1) – (5).

$$\dot{x}_c = v_c \cos \phi_c \quad (1)$$

$$\dot{y}_c = v_c \sin \phi_c \quad (2)$$

$$\dot{\phi}_c = \omega_c \quad (3)$$

$$v_c = \frac{r_w(\omega_r + \omega_l)}{2} \quad (4)$$

$$\omega_c = \frac{r_w(\omega_r - \omega_l)}{d_w} \quad (5)$$

These were extended by identifying the positions of the left and right wheels with axis on the center of gravity expressed by equations (6) – (9).

$$x_{pl} = x_c - \frac{d_w}{2} \sin \phi_c \quad (6)$$

$$y_{pl} = y_c + \frac{d_w}{2} \cos \phi_c \quad (7)$$

$$x_{pr} = x_c + \frac{d_w}{2} \sin \phi_c \quad (8)$$

$$y_{pr} = y_c - \frac{d_w}{2} \cos \phi_c \quad (9)$$

With the velocity representation of the base and wheel kinematic equations, the constraints of the non-holonomic system are expressed as equations (10a) – (10b) [13].

$$\wedge(q)\dot{q} = 0 \quad (10a)$$

$$\begin{bmatrix} -\sin\phi_c & \cos\phi_c & 0 & 0 & 0 \\ \cos\phi_c & \sin\phi_c & \frac{d_w}{2} & -r_w & 0 \\ \cos\phi_c & \sin\phi_c & -\frac{d_w}{2} & 0 & -r_w \end{bmatrix} \begin{bmatrix} \dot{x}_c \\ \dot{y}_c \\ \dot{\phi}_c \\ \dot{g}_r \\ \dot{g}_l \end{bmatrix} = 0 \tag{10b}$$

Computing the dynamic equations using Lagrange formula as expressed in equation (11) and eliminating the coefficient of the Lagrange multiplier $\lambda^T(q)$, a generalized mathematical model for a differential drive robot in terms of left and right wheel is obtained as equation (12a) – (12c).

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\alpha}_i} \right) - \frac{\partial L}{\partial \alpha_i} = B\tau_i - \lambda^T(q)\lambda \tag{11}$$

$$I = \left(I_b + 2m_w \left(\frac{d_w}{2} \right)^2 \right) \tag{12a}$$

$$m = (m_b + 2m_w) \tag{12b}$$

$$\begin{bmatrix} \left(\frac{r_w}{2d_w} \right)^2 (d_w^2 m + 4I) + I_w & \left(\frac{r_w}{2d_w} \right)^2 (d_w^2 m - 4I) \\ \left(\frac{r_w}{2d_w} \right)^2 (d_w^2 m - 4I) & \left(\frac{r_w}{2d_w} \right)^2 (d_w^2 m + 4I) + I_w \end{bmatrix} \begin{bmatrix} \ddot{g}_r \\ \ddot{g}_l \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \tau_r \\ \tau_l \end{bmatrix} \tag{12c}$$

Where the expressions are in relations to Figure 1.

This expression can be extended to differential drive robots, having the wheel axis, some distance “d” away from the center of gravity and supported by a castor wheel. The positions of the left and right wheels can be expressed by equations (13) – (16).

$$x_{pl} = x_{pb} - \frac{d_w}{2} \sin\phi_c \tag{13}$$

$$y_{pl} = y_{pb} + \frac{d_w}{2} \cos\phi_c \tag{14}$$

$$x_{pr} = x_{pb} + \frac{d_w}{2} \sin\phi_c \tag{15}$$

$$y_{pr} = y_{pb} - \frac{d_w}{2} \cos\phi_c \tag{16}$$

Where the new wheel axis is located based on equations (17) – (18).

$$x_{pb} = x_c - d \cos\phi_c \tag{17}$$

$$y_{pb} = y_c - d \sin\phi_c \tag{18}$$

However, a system where the wheel axis aligns with the center of gravity expressed equation (12c) is used.

2.2. SolidWorks CAD modelling

In order, to transform the conceptualized physical model into a visualization model capable of interfacing platforms, for example, from the CAD software to MATLAB software for further analysis, SolidWorks Premium 2014 software was used. Figure 3 illustrates the assembly of the different parts that makes a differential drive robot. With the intention of placing a robotic arm on the wooden base for future study, iron support with bearing is placed on the wooden base.

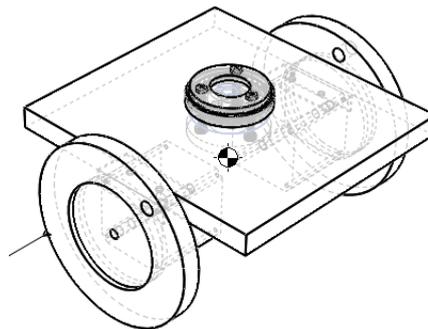


Fig. 3. SolidWorks visualization of differential drive robot without controls.

Each ‘part’ of the assembly is assigned a material type to depict a real-life system. The assembly is further constrained in 3D space using the ‘mate’ function. The motor actuator shafts are further constrained to be floating in order, to carry out motion analysis on them. For control simulation, motion study is required on the model. This involves selecting motion analysis menu where the motion profile of time: 5 s, motor type: rotary motor actuator and speed: 25 RPM are selected for the actuator shafts with attached wheels. The model is finally exported to MATLAB via the Simscape Multibody link Add-Ins where STEP for First Generation (1G) or STL files are generated on MATLAB interface.

2.3. MATLAB/ Simulink modelling

2.3.1. Importing SolidWorks file

To import and load the STL files on MATLAB, the SolidWorks Add-On must be installed. Using MATLAB Professional R2017a (64-bit), the STL file was imported using `mech_import(‘DiffDRobotie.xml’)` which generate subfolders. On running the Simulink file, Figure 4 is generated which represents the individual ‘part’ blocks and relationship with other ‘parts’ with respect to the types of joints (revolute or prismatic) and the environment. After running the overall model, Figure 5 is obtained which is a MATLAB/ Simulink visualization of the Differential Drive Robot.

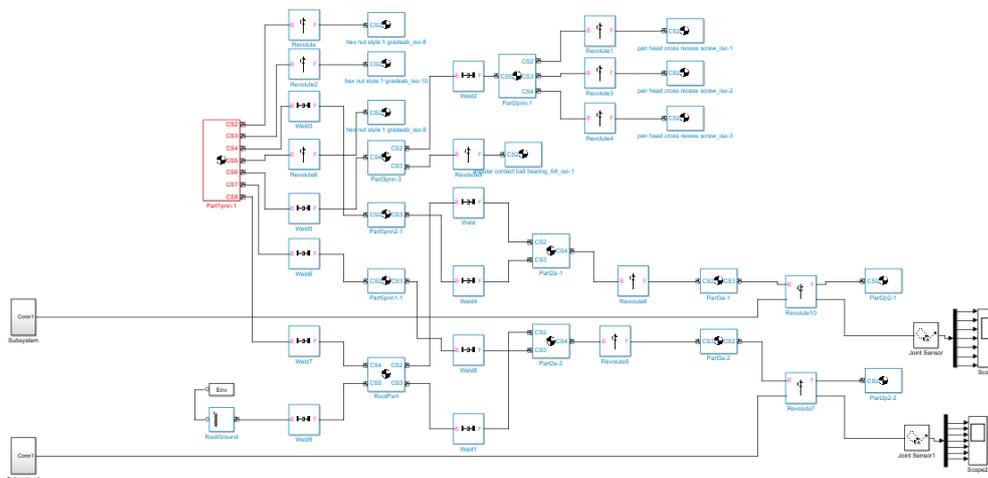


Fig. 4. MATLAB ‘part’ blocks of a differential drive robot with slight control modification on the wheels.

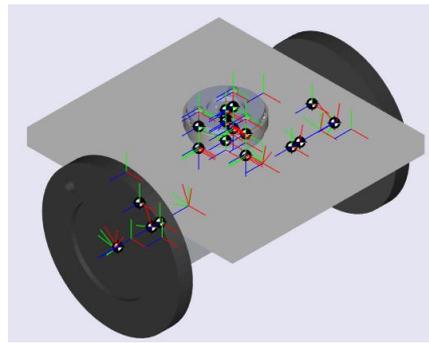


Fig. 5. MATLAB/ Simulink visualization of differential drive robot with control modification.

2.3.2. Joint actuation

The imported block serves as the plant or process model for a control system as shown in Figure 7. Although in Figure 4, a pre-actuation control or test signal was introduced at the shaft-wheel revolute joint to cause a dynamic motion of the wheel shown in Figure 6. This describes a differentiable test signal to ensure continuous-time function of the system. It inputs the angular position, speed and acceleration into the revolute joints.

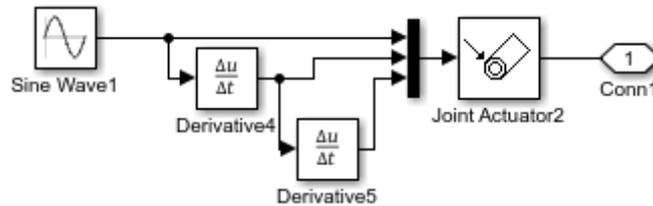


Fig. 6. Pre-actuation control or test signal.

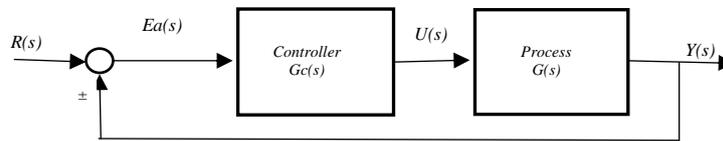


Fig. 7. Basic control system block with unity feedback [14].

From Figure 2, stepper motor actuators are used which introduces torques (τ_r, τ_l) in relations to equation (12c). Wooden wheels of weight 0.1 kg and diameter 10 cm are also used to allow mobility of the robot which will be replaced by real wheels of mass (m_w) . The dynamics of the stepper motor is summarized by equation (19a) – (19b) [4].

$$\frac{d}{dt} \begin{bmatrix} i_{as} \\ i_{bs} \\ \omega_{rm} \end{bmatrix} = \begin{bmatrix} -\frac{r_s}{L_{ss}} & 0 & \frac{RT\psi_m}{L_{ss}} \sin(RT\theta_{rm}) \\ 0 & -\frac{r_s}{L_{ss}} & -\frac{RT\psi_m}{L_{ss}} \cos(RT\theta_{rm}) \\ -\frac{RT\psi_m}{J} \sin(RT\theta_{rm}) & \frac{RT\psi_m}{J} \cos(RT\theta_{rm}) & -\frac{B_m}{J} \end{bmatrix} \begin{bmatrix} i_{as} \\ i_{bs} \\ \omega_{rm} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ -\frac{T_L}{J} \end{bmatrix} + \begin{bmatrix} \frac{1}{L_{ss}} & 0 \\ 0 & \frac{1}{L_{ss}} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} U_{as} \\ U_{bs} \end{bmatrix} \quad (19a)$$

$$T_L = T_d \sin(4RT\theta_{rm}) \quad (19b)$$

In order, to actuate a stepper motor, a microcontroller (Arduino) and a driver circuit (A4988) are required. Using the Simscape toolbox on Simulink Figure 8 illustrates the actuation of a stepper or bipolar motor.

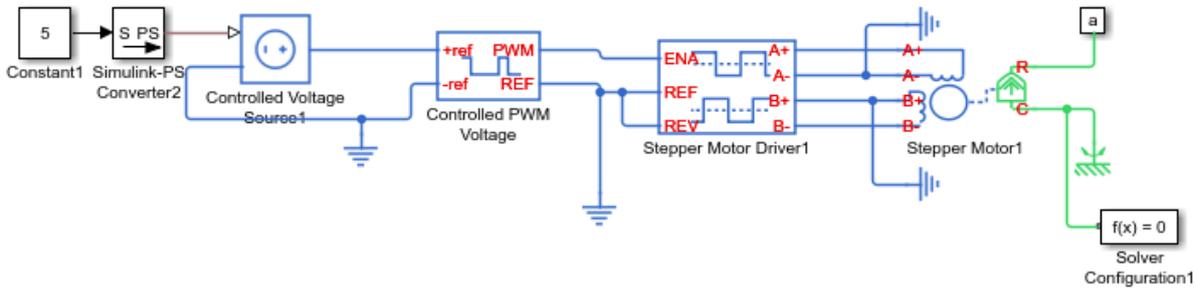


Fig. 8. Simscape simulation of a bipolar stepper motor.

The required torque per motor can be estimated according to equations (20a) – (20b).

$$\tau_r = K_T \times f(i_{as1}, i_{bs1}) \tag{20a}$$

$$\tau_l = K_T \times f(i_{as2}, i_{bs2}) \tag{20b}$$

The function of the microcontroller extends beyond generating a pulse signal (PWM) to actuate the motor driver, it also serves as the overall system controller needed for the physical model coordination.

2.3.3. Introducing controls to the system

Introducing the self-balancing concept of wheeled robot [4, 8, 15] with dynamics shown in equation (21) placed in cascade with the differential drive robot dynamics, the tilt angle of the robot is observed as θ and with an accelerometer sensor a feedback signal is obtained.

$$\begin{bmatrix} 2 \left(m_w + \frac{I_w}{r_w^2} + \frac{1}{2} m_b \right) & -m_b l \cos \theta \\ -m_b l \cos \theta & (l^2 m_b + I_b) \end{bmatrix} \begin{bmatrix} \ddot{x} \\ \ddot{\theta} \end{bmatrix} + \begin{bmatrix} m_b l \dot{\theta}^2 \sin \theta \\ -m_b g l \sin \theta \end{bmatrix} = \begin{bmatrix} \frac{1}{r_w} (\tau_r + \tau_l) \\ 0 \end{bmatrix} \tag{21}$$

The LQR control scheme was deployed to maintain the tilt angle at zero degrees. This adopts the quadratic performance index concept expressed in equation (23a). First, the dynamics were reduced to a first-order differential equation, linearized about a steady-state after perturbing the system, then with Taylor series expansion expressed in equation (22a) – (22b), the linear dynamics was obtained.

$$f(P, Q, \dots) = f(P^*, Q^*, \dots) + \left. \frac{\partial f}{\partial P} \right|_{\substack{P=P^* \\ Q=Q^*}} \delta P + \left. \frac{\partial f}{\partial Q} \right|_{\substack{P=P^* \\ Q=Q^*}} \delta Q + \dots \tag{22a}$$

$$\Delta f(P, Q, \dots) = f(P, Q, \dots) - f(P^*, Q^*, \dots) \tag{22b}$$

The scheme helps to obtain an optimum feedback gain described by the Riccati matrix described by equation (23b) [16].

$$J = \int_{t_0}^{t_1} (X^T Q X + U^T R U) \delta x \tag{23a}$$

$$U_{opt} = -R^{-1} B^T P X = -K X \tag{23b}$$

where P is a square symmetric matrix and K is the feedback gain.

3. RESULTS AND DISCUSSION

Figure 9 depict the result obtained for motion study for a period of 5 seconds, which corresponds with the output of Figure 5 when a sine wave signal (with differentiable profile) in Figure 4 was used to actuate the revolute joint at the two wheels.

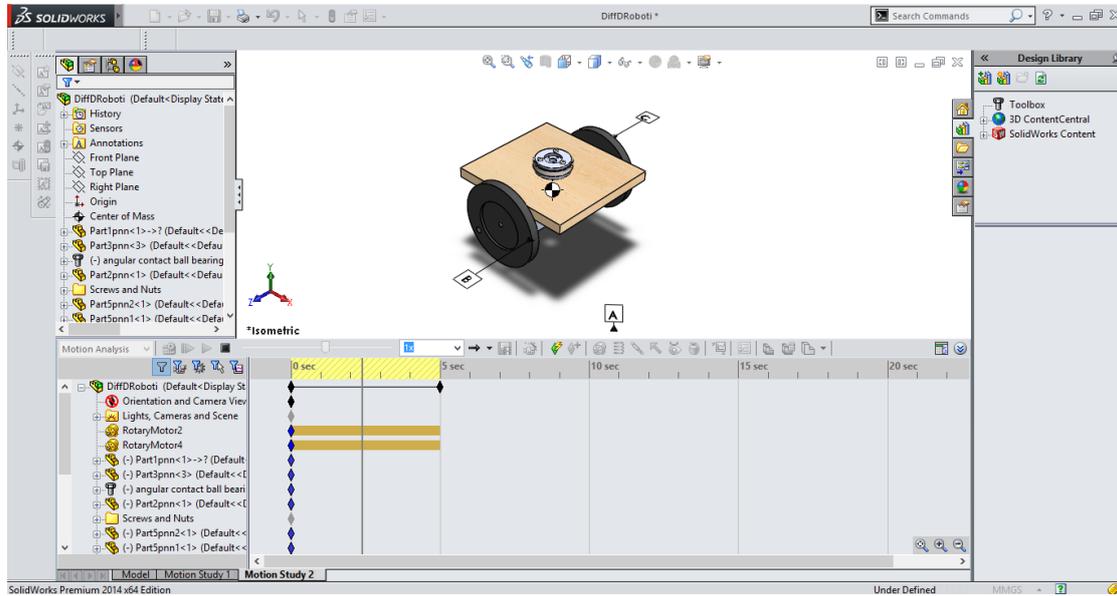


Fig. 9. Motion study on SolidWorks software.

The output from Figure 4 (SolidWorks transformed to MATLAB/Simulink) illustrates the angular position (rad), velocity (rad/s), acceleration (rad/s²), computed torque (Nm), reaction torque (Nm) and reaction force (N) with respect to the reaction at the base as shown in Figure 10 for the left and right wheel in response to an input sine wave. Introducing a torque sensor to Figure 8, an output torque in Nm is obtained with results shown in Figure 11.

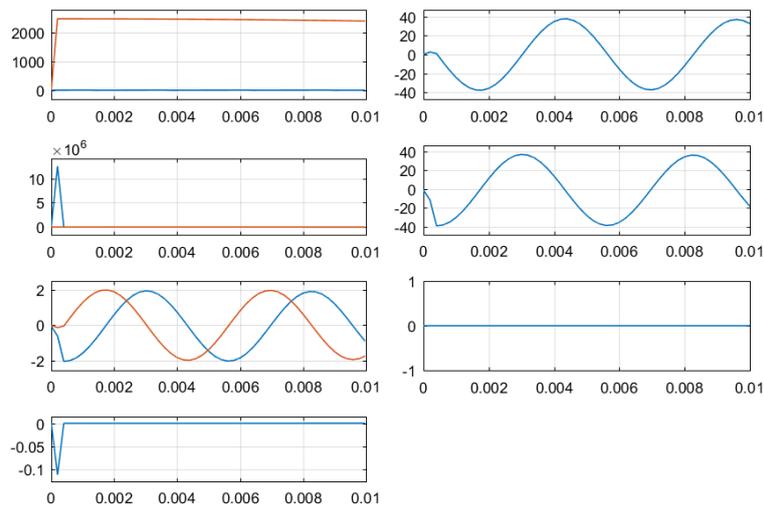


Fig. 10. Output of SolidWorks transformed to MATLAB/Simulink model.

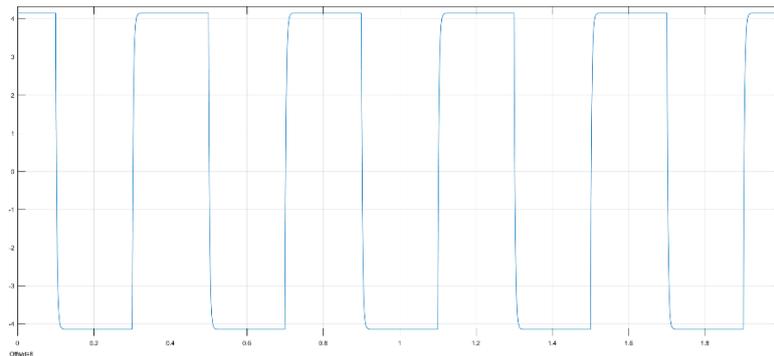


Fig. 11. Output torque in Nm of a Bipolar stepper motor.

The angular speed obtained from a bipolar stepper motor is shown in Figure 12 in response to a PWM control input from a microcontroller. The microcontroller requires written codes (instruction codes) in C language, inputted and compiled on an Arduino IDE software, and the output of the code is after successful compilation is upload to the real-time microcontroller.

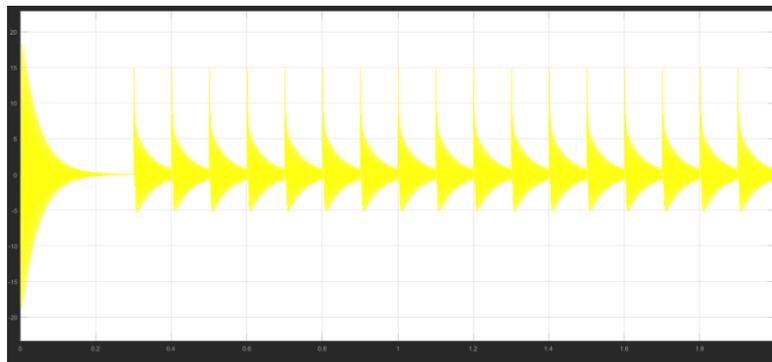


Fig. 12. The angular speed ω in rad/sec for the left and right bipolar stepper motors.

The instruction code contains the controller optimum gains obtained from the LQR control scheme to set the tilt angle to zero in relations to the vertical axis and control inputs to actuate bipolar stepper motor specifying the steps and direction pins. Once the compiler successfully runs the program code an equivalent machine code (hex file) is generated. This is then uploaded to the microcontroller via an external control mode.

4. CONCLUSIONS

This paper has given a modular design concept of a mobile robot which involves importing a modelled system from SolidWorks into MATLAB/Simulink and validating the system with an equivalent mathematical model. The joint actuation was observed with respect to a smooth differentiable sine wave and a modelled bipolar stepper motor. Experimental analysis was carried out involving an interfaced Simulink with an emulator (Arduino Real-Time Operating System). Parameter estimation toolbox on Simulink also helped to penalise a developed cost function to match the physical model to an experimental model. The LQR control scheme was further used to establish the stability of the system and maintain the tilt angle to zero degrees using the self-balancing model estimates.

REFERENCES

- [1] Keuchel, U., Stephan, R.M., Microcontroller-based adaptive control applied to thyristor-driven dc-motors, Advances in Industrial Control, Springer-Verlag London, 1994.
- [2] Moe, S., Guidance and control of robot manipulators and autonomous marine robots, PhD Thesis Cybernetics Engineering, Faculty of Information Technology, Mathematics and Electrical Engineering, Norwegian University of Science and Technology, 2016.

- [3] Chase, J., How AI changes the rules: new imperatives for the intelligent organization, Report MIT SMR CONNECTIO, Custom Research Report, Massachusetts Institute of Technology, 2020, p. 1-21.
- [4] Owojori, A.O., Akingbade, K.F., Apena, W.O., Ogunti, E.O., Modelling of a differential drive robot in preparation for control analysis, Proceedings of the 3rd International Conference on Electrical, Communication and Computer Engineering (ICECCE), Kuala Lumpur, Malaysia, 2021.
- [5] Villani, V., Pini, F., Leali, F., Secchi, C., Survey on human-robot collaboration in industrial settings: safety, intuitive interfaces and applications, Mechatronics Journal, 2018, p. 1-19.
- [6] Guo, Z.Y., Hsia, T.C., Joint trajectory generation for redundant robots in an environment with obstacles, Journal of Robotic Systems, vol. 10, no. 2, 1993, p. 199-215.
- [7] Lombai, F., Szederkenyi, G., Trajectory tracking control of a 6 degree of freedom robot arm using nonlinear optimization, IEEE, 2008, p. 655-660.
- [8] Bonafilia, B., Gustafsson, N., Nyman, P., Nilsson, S., Self-balancing two wheeled robot, Intelligent Robotics and Autonomous Agents series 2013, p. 1-11.
- [9] Nourbakhsh, I.R., Roland, S., Introduction to autonomous mobile robots, Intelligent Robotics and Autonomous Agents series, 2004, p. 12-45.
- [10] Olofsson, B., Nilsson, H., Robertsson, A., Akesson, J., Optimal tracking and identification of paths for industrial robots, IFAC Proceedings Volumes, 2011, p. 1126-1132.
- [11] Ibrahim, A., Alexander, R.R., Sanghar, M.S.U., D'Souza, R.D., Control systems in robotics: a review, International Journal of Engineering Inventions, vol. 5, no. 5, 2016, p. 29-38.
- [12] Pachaiyappan, S., Micheal, M., Sridhar, T., Design and analysis of an articulated robot arm for various industrial applications, IOSR Journal of Mechanical and Civil Engineering (IOSR-JMCE), 2014, p. 42-53.
- [13] Seegmiller, N.A., Dynamic model formulation and calibration for wheeled mobile robots, Doctorate Thesis, The Robotics Institute Carnegie Mellon University, 2014.
- [14] Bishop, R.H., Modern control systems analysis and design using MATLAB, Addison-Wesley Publishing Company, 1997.
- [15] Jamil, O., Jamil, M., Ayaz, Y., Ahmad, K., Modeling, control of a two-wheeled self-balancing robot, International Conference on Robotics and Emerging Allied Technologies in Engineering (iCREATE), Islamabad, Pakistan, 2014, p. 191-199.
- [16] Hellman, H., Sunnerman, H., Two wheeled self-balancing robot: design and control based on the concept of an inverted pendulum, Bachelor's Thesis in Mechatronics, 2015.