



## Simulation of a Self-Balancing Platform on the Mobile Car

Bushra Amer Tawfeeq\*

Maher Yahya Salloom\*\*

Ahmed Alkamachi\*\*\*

\*, \*\*, \*\*\* Department of Mechatronics engineering/ Al-Khwarizmi Engineering College/ University of Baghdad

\*Email: [bushra.a.tawfeeq@gmail.com](mailto:bushra.a.tawfeeq@gmail.com)

\*\*Email: [drmahir@kecbu.uobaghdad.edu.iq](mailto:drmahir@kecbu.uobaghdad.edu.iq)

\*\*\*Email: [ahmed78@kecbu.uobaghdad.edu.iq](mailto:ahmed78@kecbu.uobaghdad.edu.iq)

(Received 27 July 2021; Revised 17 September 2021; Accepted 29 September 2021)

<https://doi.org/10.22153/kej.2021.09.003>

### Abstract

In the last years, the self-balancing platform has become one of the most common candidates to use in many applications such as flight, biomedical fields, industry. This paper introduced the simulated model of a proposed self-balancing platform that described the self-balancing attitude in (X-axis, Y-axis, or both axis) under the influence of road disturbance. To simulate the self-balanced platform's performance during the tilt, an integration between Solidworks, Simscape, and Simulink toolboxes in MATLAB was used. The platform's dynamic model was drawn in SolidWorks and exported as a STEP file used in the Simscape Multibody environment. The system is controlled using the proportional-integral-derivative (PID) controller to maintain the platform leveled and compensate for any road disturbances. Several road disturbances scenarios were designed in the x-axis, y-axis, or both axis (the pitch and roll angles) to examine the controller effectiveness. The simulation results indicate that that the platform completed self-balancing under the effect of disturbance ( $10^\circ$  and  $-10^\circ$ ) on the X-axis, Y-axis, and both axes in less than two milliseconds. Therefore, a proposed self-balancing platform's simulated model has a high self-balancing accuracy and meets operational requirements despite its simple design.

**Keywords:** Self-balancing platform, (PID) controller, Solid work, software, MATLAB, Simscape Multibody, package.

### 1. Introduction

The self-balancing platform has always been a very challenging area to work on. It was elevated to a new level by introducing modern technologies, such as a PID controller, an IMU sensor (gyroscope and accelerometer), model simulation [1]. Self-balancing platform has witnessed a change in the shape and size depending on: the degree of freedom (tilt direction), driving mechanism, Control system, the type of internal measurement unit(IMU), supplement machinery size (whether it is a cart, or agricultural machinery, or a simple robot smart car.), and weight of the load [2]. The compensation for the platform's angular fluctuation during its movement in the uneven

ground makes measuring the platform's tilt angle by a potentiometer or more advanced sensors more necessary. The obvious choice for compensating the platform's tilt is motor, such as servo motors, stepper motors, and DC motors [3].

MathWorks's mechanism to design, model, simulate, and result-validate any physical or mechanical system is the Simscape Multibody Library [4][5]. Simscape Multibody software is developed so that its block libraries are used to simulate the mechanical system according to basic physical principles (Newtonian dynamic of forces and torque). In contrast, Simulink blocks symbolize the system's mathematical operations or the processes on the signals. Still, Simscape Multibody software works in the Simulink environment and interface with Simulink and



Matlab flawlessly [6][7]. The Simulink and Simscape software has been used to build and control the inverted pendulum system without the need to derive a complicated system of mathematical differential equations [8].

The experimental results of the 2DOF self-balancing platform are shown the presence of the PID controller's fine-tuned coefficients limiting the overshoot and settling time and no steady-state error. The coefficient is analyzed by the simulation aspects (Matlab) and used to maintain the platform at an initially nominated angle when the structure's tilt is in the x-axis or y-axis (the pitch and roll angles). The two servo motors depend on the PID controller's feedback to support and balance the top wood platform through tilt [9]. Self-balancing platform with 2DOF on a cart simulated model relies upon MatlabSimMechanics and Simulink of DC motor. The model was configured as an inverted pendulum conception in a square-shaped with each side measuring 30 cm attached with the three links that covered all required support of the platform. Satisfactory performance of the existing platform, stability, re-correct timeless than 1second, and the highest level of precision in determining the servomotor's next position is among the ones that have been attributed to using the MatlabSimMechanics simulation and Simulink of DC motor [10].

Platform's SimMechanics simulation has been a fundamental parameter indicating the relationship between the platform's navigation parameters (roll, yaw, and pitch angles) and selecting the PID coefficient regarding the servomotor's rotation angle. Also, the simulation named the most critical limits that should exceed in the design as the platform contained the Arduino UNO control the servomotors through three links to regulate the platform's tilt [11]. In terms of meeting the top plate in the 6 DOF platform, namely, vehicle stabilizer, balance requirements when there is an alter of the base plate's position. The Simscape simulation has been added. The vehicle stabilizer's simulated model had involved six linear actuators connected to the base plate and top plate with universal

joints. The Simscape simulation's result of vehicle stabilizer has been assessed; this leads to the conclusion that the PD controller's availability to select linear actuators' desired position is adequate to stabilize the upper platform's position, especially at large angles (more than 45° on all axis) [12].

The displacement curve and the force curves of six hydraulic cylinders of the 6DOF hydraulic platform are extracted from kinematic simulation (Matlab) and dynamic modelling (SolidWorks/Adams software). The displacement curve application gave good results that helped the platform's hydraulic cylinders run smoothly and continuously during the self-levelling. The dynamic modelling is used to select suitable materials for (upper and lower planes, hydraulic cylinders, hinge circuits) to hold out the cylinder's maximum extension force and the maximum binding force of the ball hinge [13]. A Genetic Algorithm(GA)is a global algorithm base on Darwin's theory of evolution. Genetic algorithm is one of the PID controller optimization methods, especially if the PID controller interconnected system is a DC motor [14]. Accordingly, the genetic algorithm often provides a reasonable balance between the PID controller coefficient's accurate values and better PID response regarding the rise time and the settling time. GA consists of four main stages (initial generation, cost function, selection techniques, Crossover, and mutation [15] [16] [17].

This study's main contribution is Use of the SolidWorks and Simscape environments to build, model, and control the proposed self-balancing platform's simulated model without the need to derive a complex system of mathematical differential equations.

## 2. Methodology

The simulated model of the proposed self-balancing platform on the mobile car has been built based on the working principle of the block diagram shown in figure.1.

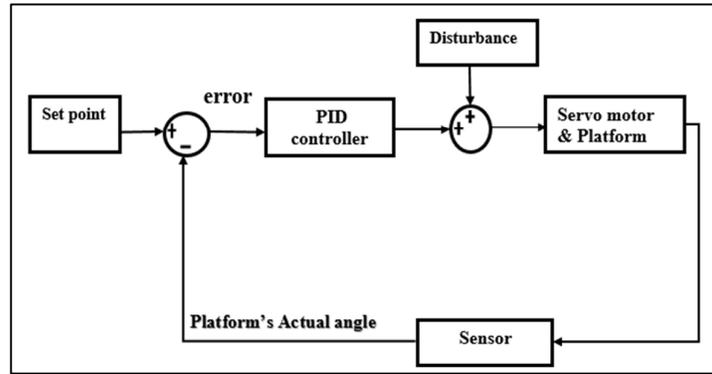


Fig. 1. The block diagram of Simscape / Simulink model of self-balancing platform on the mobile car.

### 3. Simscape / Simulink model of the proposed self-balancing platform on the mobile car

The Simscape / Simulink model of the proposed self-balancing platform on the mobile car, consisting of five main Simulink Subsystems. The Subsystems in the simulated model of the self-balancing platform on the mobile car, except Platform Subsystem, have been designed by (MATLAB R2020a) / Simulink. The platform subsystem was designed by SolidWorks(SW2020), then imported to MATLAB/ Simulink through the MatlabSimscape Multibody package / (MATLAB R2020a).

#### 3. 1. Input parameters subsystem

The input parameters participants in the simulated model of the self-balancing platform on the mobile car (for X-axis /middle plate or Y-axis / top plate) were divided into two different inputs; one of the inputs defined as the set point of roll or pitch angles ( $0^\circ = 0$  radians) and entered by a (MATLAB R2020a)/ Simulink Library Browser / constant value block to PID controller of the respective plate. The second input mimics a road disturbances signal (tilt angles (in degree) in the X-axis or Y-axis, or both axes), designed by a (MATLAB R2020a) / Simulink Library Browser/ Signal Builder block. The degrees to radian block converted the tilt angles of disturbances signal to radian before entering the plate's PID controller. The input parameters subsystem is shown in figure.2.

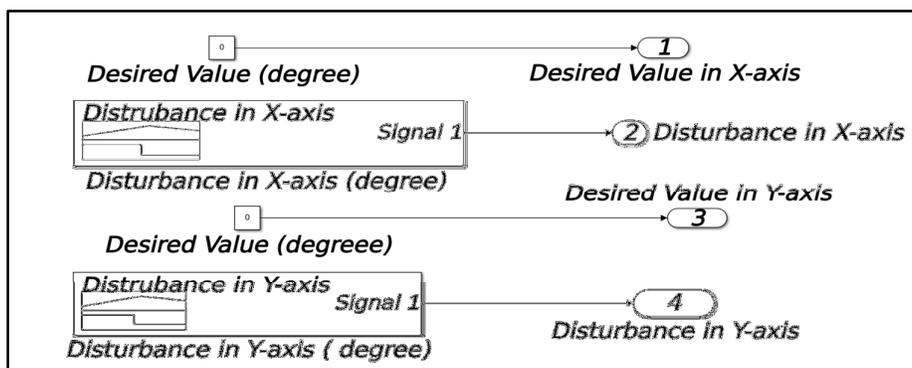


Fig. 2. Input parameters Subsystem.

#### 3.2. PID Controller Subsystem

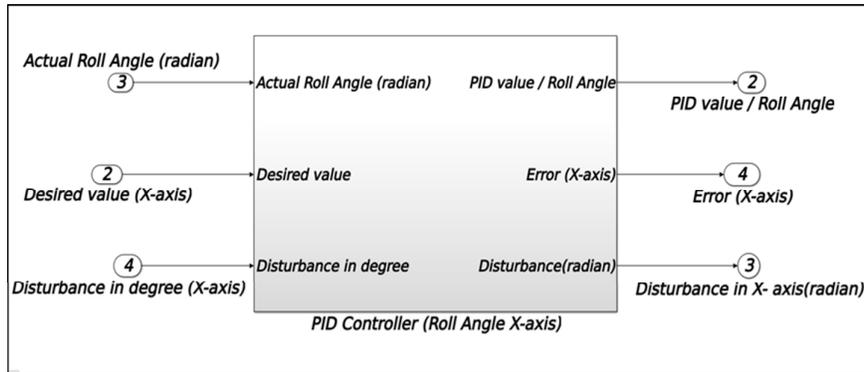
According to the mathematical modeling of Eq. (1) [18] [19], the two PID controllers of the simulated model for the self-balancing platform on the mobile car have been established by

MATLAB (R2020a) / Simulink. PID controller Subsystem contained two controllers; one PID controller is for the Pitch Angle Y-axis (top late), the other PID controller for the Roll Angle X-axis (middle plate) as shown in figure.3.

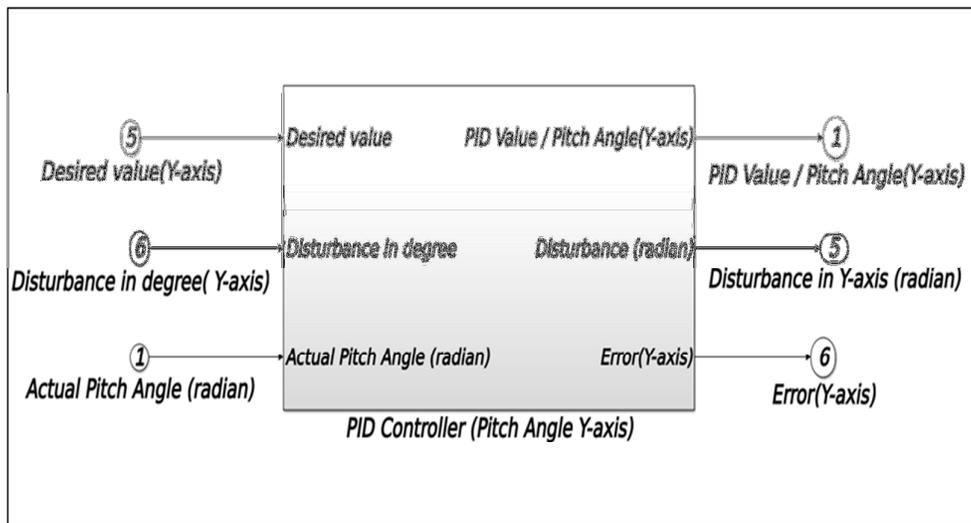
$$u(t) = K_p \times e(t) + K_i \int_0^t e(t)dt + K_d \frac{de}{dt} \dots (1)$$

Where:  
 Kp: the proportional value.

Ki: the integral value.  
 Kd: the derivative value; et is the error function; t is the Time.



A. PID Controller for Roll Angle X-axis.



B. PID Controller Pitch Angle Y-axis.

Fig. 3. PID Controller Subsystem.

### 3.3 DC Servo motor subsystem

The motor's mathematical model describes the motor's performance using mathematical equations [20] [21]. The following mathematical modelling and kinematics modelling formulas have been used with MATLAB R2020a /Simulink's functionality to dispensing the DC servo motor's intricate design by SolidWorks and replace it with a simple simulated model. This method has been used for both DC servo motors of the simulated model for the self-balancing platform on the mobile car. DC servo motor Subsystem contained two DC servo motors, one

of the motors is for the Pitch Angle Y-axis (top plate), the other motor is for the Roll Angle X-axis (middle plate), as shown in the figure. 4. The kinematics modelling of DC servo motor for both Pitch Angle Y-axis Servo motor and the Roll Angle X-axis Servo motor (shown in figure.4) were simulated in Simulink as shown in Fig. figure.6. according to the equations (2) and (3) [22], the below block diagram of the DC servo motor is shown in figure.5 [23] and the motor parameters in the Table1. For this simulation, the parameters of an actual DC Servo motor were chosen from [19] and shown in Table1.

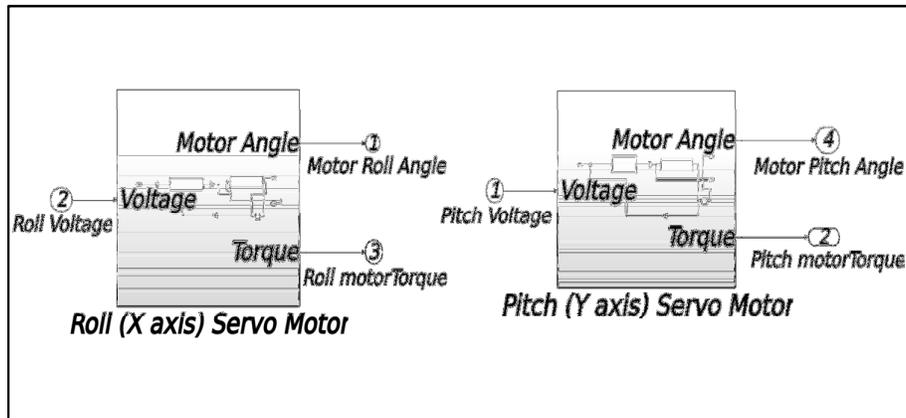


Fig. 4. DC servo motor Subsystem.

$$G_{position}(s) = \frac{\theta(s)}{V(s)} = \left[ \frac{K_m}{(L_a s + R_a)(Js + B) + K_b K_m} \right] \frac{1}{s} \quad \dots (2)$$

$$G_{speed}(s) = \frac{\dot{\theta}(s)}{V(s)} = \frac{K_m}{(L_a s + R_a)(Js + B) + K_b K_m} \quad \dots (3)$$

Where:

- $R_a$  : Armature resistance, ohm.
- $L_a$  : Armature inductance, henry.
- $i_a$  : Armature current, ampere.
- $V_a$  : Armature voltage, volts.
- $V_b$  : Back EMF, volts.

- $K_b$  : Back EMF constant, volt/(rad/sec).
- $K_m$  : Torque constant, N-m/Ampere.
- $T_m$  : Torque developed by the motor, N-m.
- $T_l$  : Torque developed by the load, N-m.
- $T_d$  : Disturbance torque, N-m.
- $\theta(t)$ : Angular displacement of shaft, radians.
- $J$  : Moment of inertia of the motor and load, Kg-m<sup>2</sup> /rad.
- $B$ : Frictional constant of motor and load, N-m/(rad/sec).

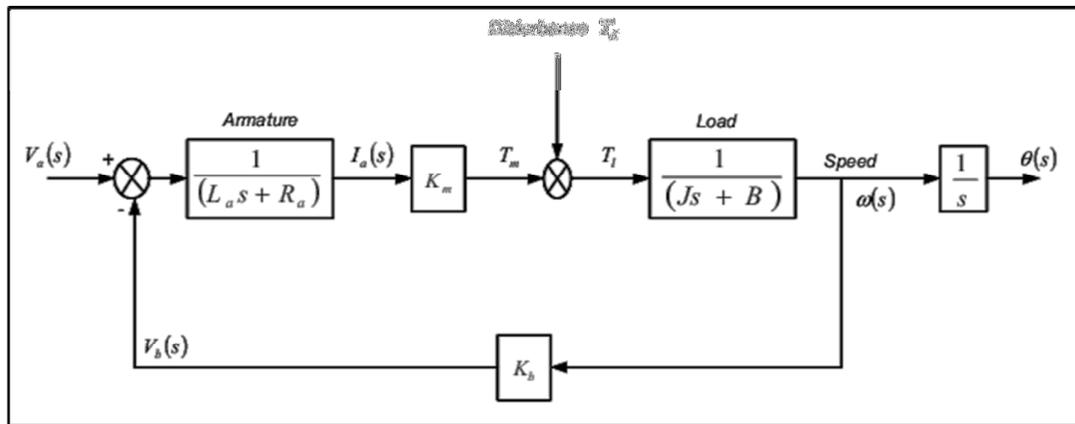


Fig. 5. Block diagram of DC servo motor [23].

Table 1, Actual DC Servo Motor Parameters [23].

S.No.	Parameters	Values
1	Armature Resistance ( $R_a$ )	1.17 ohm
2	Armature inductance ( $L_a$ )	0.5H
3	coulomb friction torque (B)	0.01 N-m/(rad/sec)
4	Rotor inertia (J)	0.01 Kg- m <sup>2</sup> /rad
5	Motor torque constant ( $K_m$ )	0.1 V/rad/sec
6	Back EMF constant ( $K_b$ )	0.71 N.m/ Ampere
7	Operating voltage (V)	12V
8	Operating current (I)	2.28A
9	Maximum Torque	8Kg-cm

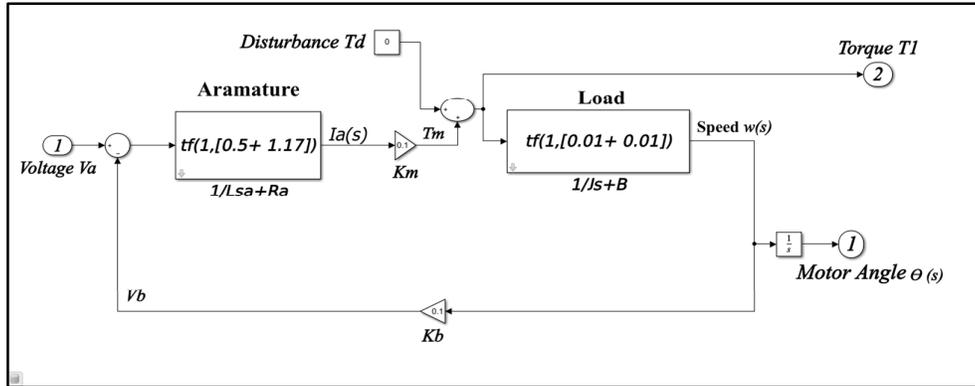


Fig. 6. Kinematics modelling of DC Servo motor using Simulink.

**3. 4. Platform subsystem**

In terms of dynamic modelling, a 3D model of the self-balanced platform on the mobile car was constructed and optimized by SolidWorks(SW2020) based on the design parameters and related limitations. The platform's complete system (three plates and two holders) consisted of lightweight material (Perspex GS cast Acrylic Glass 6mm thickness) with a dimension of (45 cm length × 30 cm width) at the rectangle base plate (40cm length× 40 cm width× 3cm inside) at the square middle plate, (20cm length×

20 cm width) the square top plate and each holder with (15cm width× 25cm height). Besides, the platform attached to a reliable iron rectangle that had the advantage of having a well-balanced structure (45cm length × 60 cm width) stands on four tires, as shown in figure.7. MatlabSimscape Multibody package (MATLAB R2020a) was used to personify the 3D SolidWorks design of the self-balancing platform on the mobile car as MATLAB Simscape model. Figure.8 illustrates the Simscape model of the self-balancing platform on the mobile car.

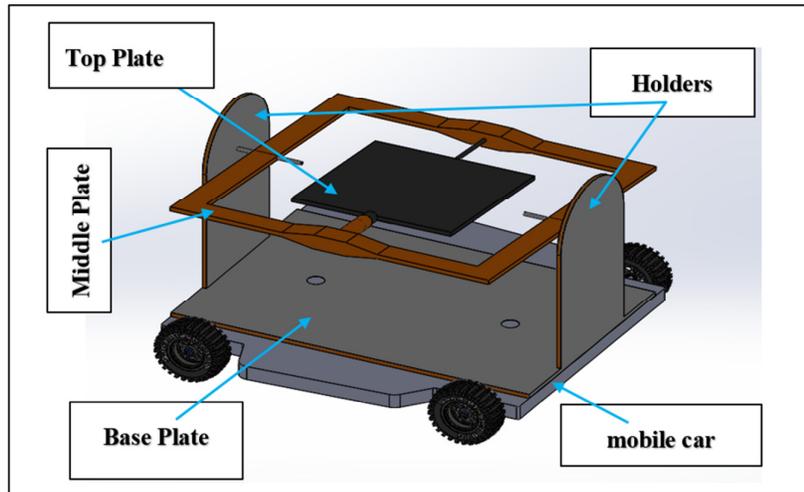


Fig. 7. The CAD drawing for the proposed prototype of the 3D self-balancing platform on mobile car.

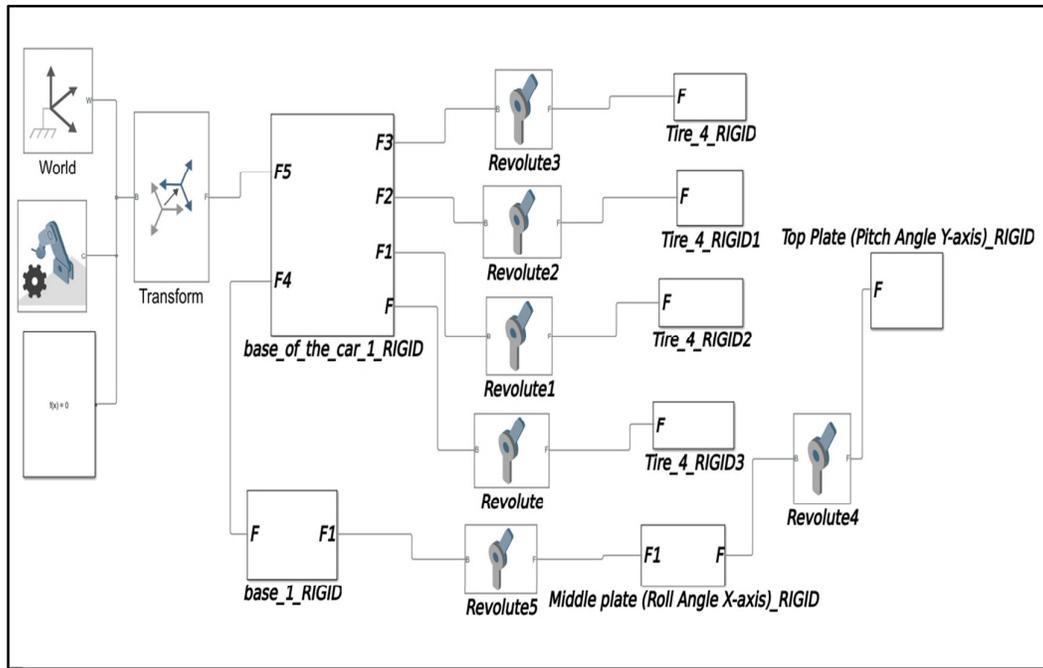


Fig. 8. The Simscape model of the self-balancing platform on the mobile car.

### 3.5. Sensor subsystem

Six revolute joints were used to connect eight rigid frames to each other due to conversion from the CAD design of the self-balancing platform on the mobile car to the Simscape model. Each revolute joint represents position relation type between platform CAD's sub-assemblies. The used technique for applying torque (N-m) to rotate the plate's (either middle or top) for self-balancing the plate's inclination is a revolute joint primitive/ Actuation properties / Actuation Torque Setting (t) [24] [25] as shown in Figure.9. The DC servo motor was simulated by MATLAB / Simulink, and the joint connection between the plate (either middle or top) and the DC servo motor is a Simscape block. Therefore, the S-PS (Simulink-Physical System) converter block is used to interface the two toolboxes with each other. The

used sensing technique for registering the plate's (middle and top) actual current angle value (Roll and Pitch angles) after self-balancing is firstly plate's actual velocity is measured (rad/sec) by a revolute joint primitive/ sensing properties / relative angular velocity option(w) [24] [25]. Then, the plate's actual velocity converts to the plate's actual angle(rad) by Simulink/ Integrator block. The joint connection between the plate (either middle or top) and DC servo motor is a Simscape block, and the Integrator block is a Simulink block. Therefore, the PS-S (Physical system - Simulink) converter block is used to interface the two toolboxes with each other. The PID controller output depends on revolute joint sensing's feedback to re-correct the tilt of the platform. The sensing technique in the sensor subsystem is shown in figure.9.

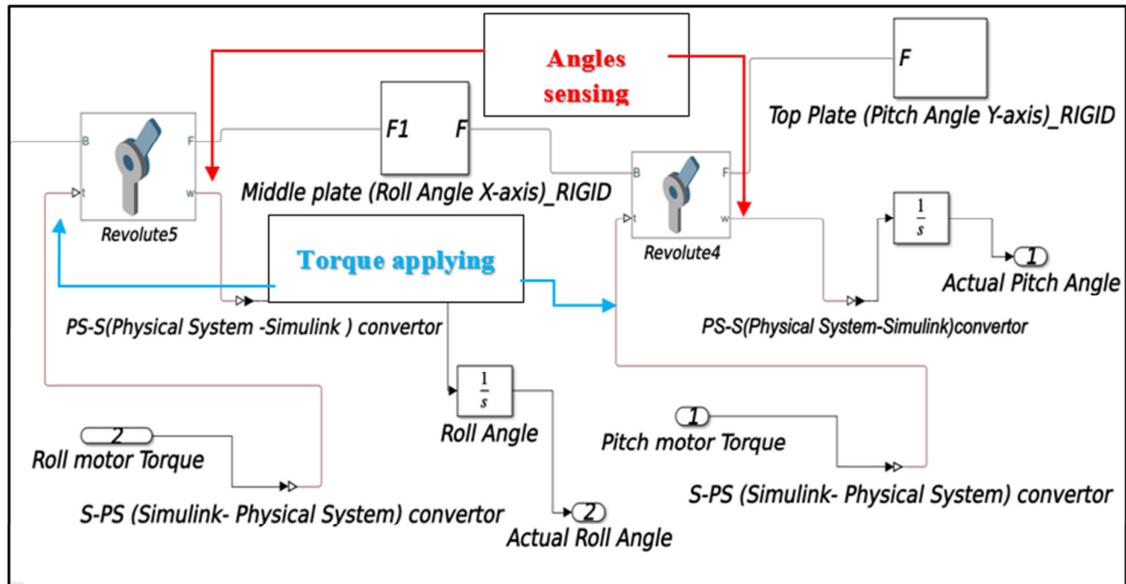


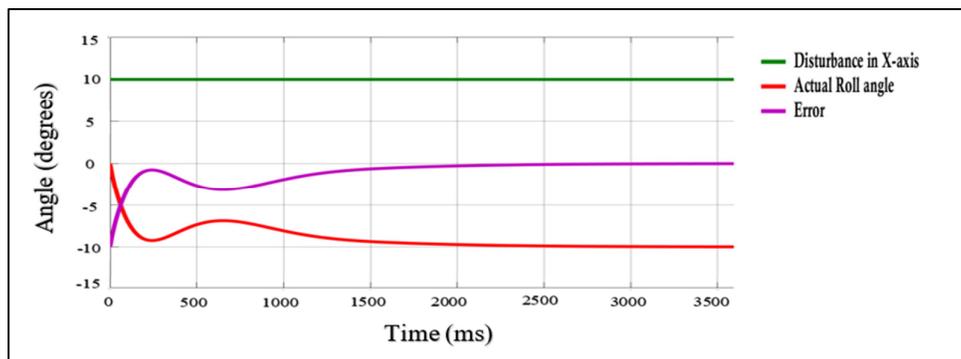
Fig. 9. Torque applying technique and angles sensing technique.

#### 4. Results and Discussion

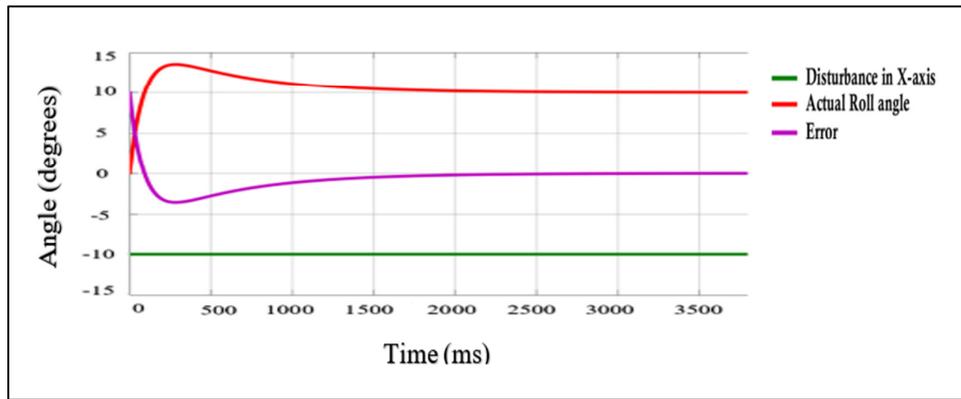
The simulated model of the self-balancing platform on the mobile car was tested in a Simulink environment included in (MATLAB R2020a) to validate simulated model capabilities to self-balancing under disturbance scenarios. The simulation results were attained by self-balancing verification of the simulated model of the self-balancing platform on the mobile car in MATLAB R2020a /Simulink. This verification was repeated four times, one for each disturbance scenario shown below in the four cases.

#### Case One: Self-balancing attitude in the simulated model under the influence of road disturbance scenario in the X-axis.

The self-balancing behavior of the simulated model for the self-balancing platform on the mobile car resulted from the platform's simulated model subjected to disturbance (tilt angle 10° on positive X-axis and -10° on negative X-axis) is shown in figure 10. The self-balancing of the (platform on the mobile car) structure in the virtual simulation environment under tilt angle 10° on positive X-axis and -10° on negative X-axis is shown in figure 11: (a, b).



A. under tilt angle 10° on positive X-axis.



B. under tilt angle  $-10^\circ$  on negative X-axis.

Fig. 10. Self-balancing behavior of simulated model on X-axis.

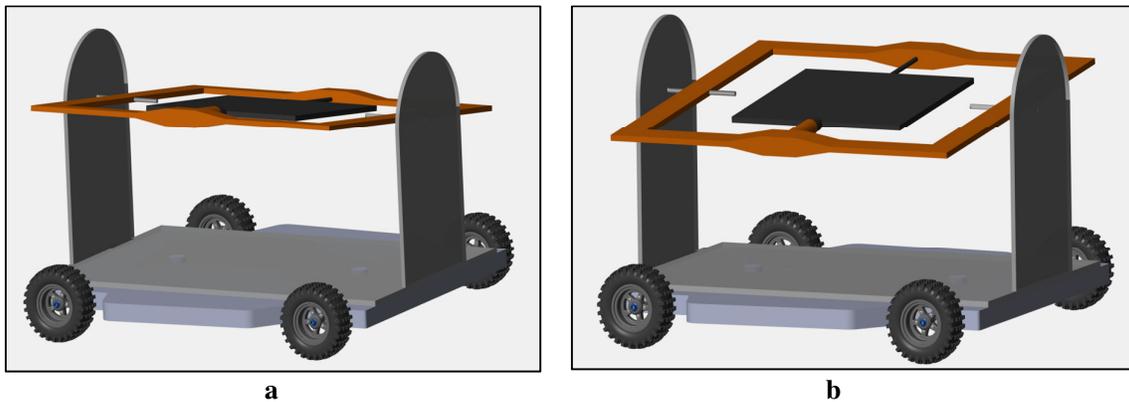
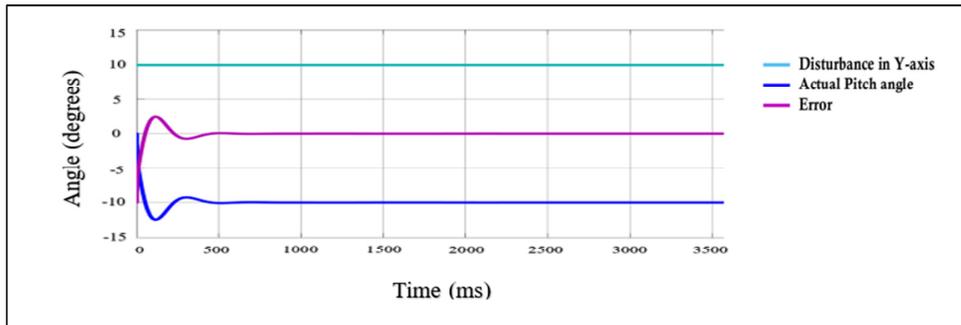


Fig. 11. Self-balancing of the (platform on the mobile car a) structure in the virtual simulation a) under tilt angle  $10^\circ$  on positive X-axis b) under tilt angle  $-10^\circ$  on negative X-axis.

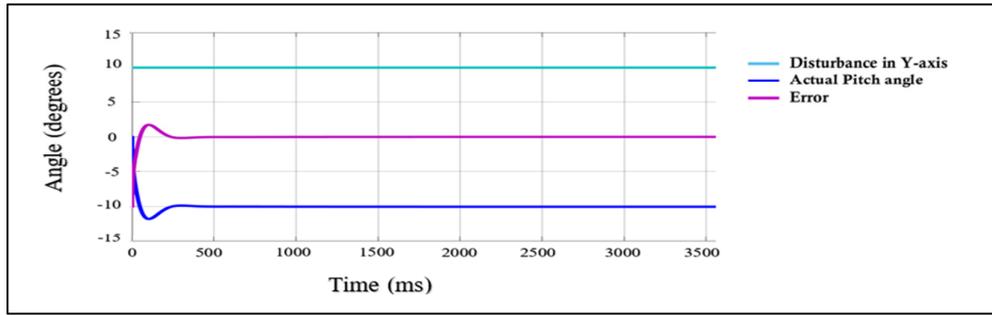
**Case Two: Self-balancing attitude in the simulated model under the influence of road disturbances scenario in the Y-axis.**

The self-balancing behavior of the simulated model for the self-balancing platform on the mobile car resulted from the platform's simulated model subjected to disturbance (tilt angle  $10^\circ$  on

positive Y-axis and  $-10^\circ$  on negative Y-axis) is shown in figure 12. The self-balancing of the (platform on the mobile car) structure in the virtual simulation environment under tilt angle  $10^\circ$  on positive Y-axis and  $-10^\circ$  on negative Y-axis is shown in figure 13: (a, b).



A. under tilt angle  $10^\circ$  on positive Y-axis.



B. under tilt angle  $-10^\circ$  on negative Y-axis.

Fig. 12. Self-balancing behavior of simulated model on Y-axis.

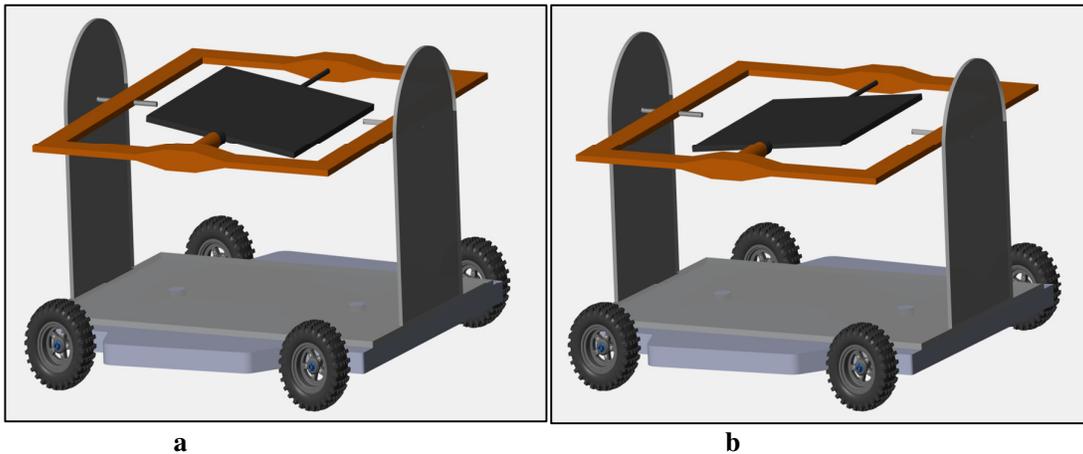
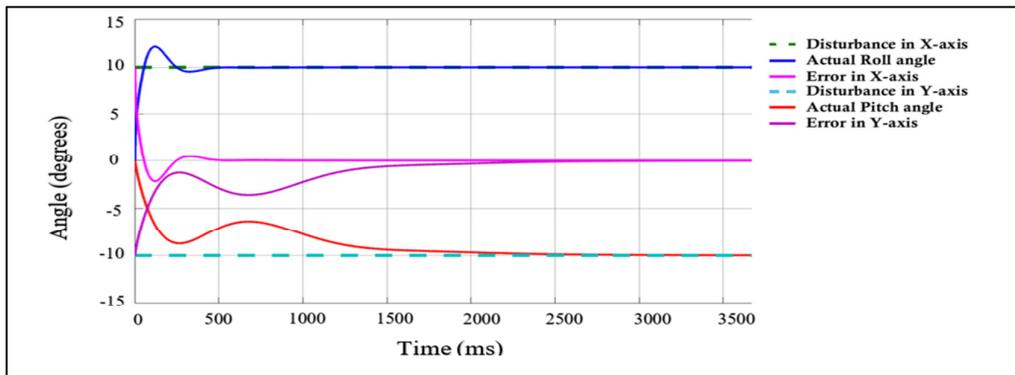


Fig. 13. Self-balancing of the (platform on the mobile car) structure in the virtual simulation a) under tilt angle  $10^\circ$  on positive Y-axis b) under tilt angle  $-10^\circ$  on negative Y-axis.

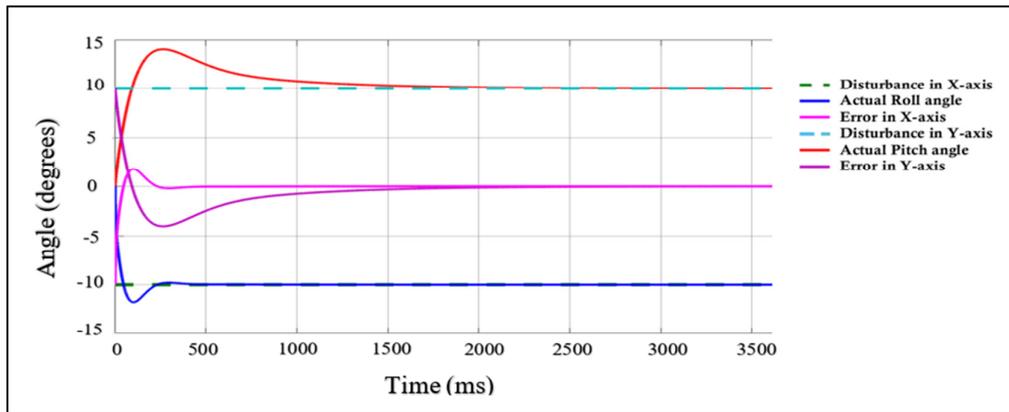
**Case Three: Self-balancing attitude of the simulated model under the influence of road disturbances scenario in the biaxial (contradictory tilt angles in X-axis and Y-axis)**

The self-balancing behavior of the simulated model for the self-balancing platform on the mobile car as a result of the platform's simulated model subjected to disturbances (tilt angle  $10^\circ$  on

positive X-axis and  $-10^\circ$  on negative Y-axis) and (tilt angle  $-10^\circ$  on negative X-axis and  $10^\circ$  on positive Y-axis) shown in figure 14. The self-balancing of the (platform on the mobile car) structure in the virtual simulation environment under tilt angle ( $10^\circ$  on positive X-axis and  $-10^\circ$  on negative Y-axis) and tilt angle ( $-10^\circ$  on negative X-axis and  $10^\circ$  on positive Y-axis) is shown in figure 15: (a, b).



A. under tilt angle ( $10^\circ$  on positive X-axis and  $-10^\circ$  on negative Y-axis).



B. under tilt angle (-10° on negative X-axis and 10° on positive Y-axis).

Fig. 14. Self-balancing behavior of simulated model on (contradictory tilt angles in X-axis and Y-axis).

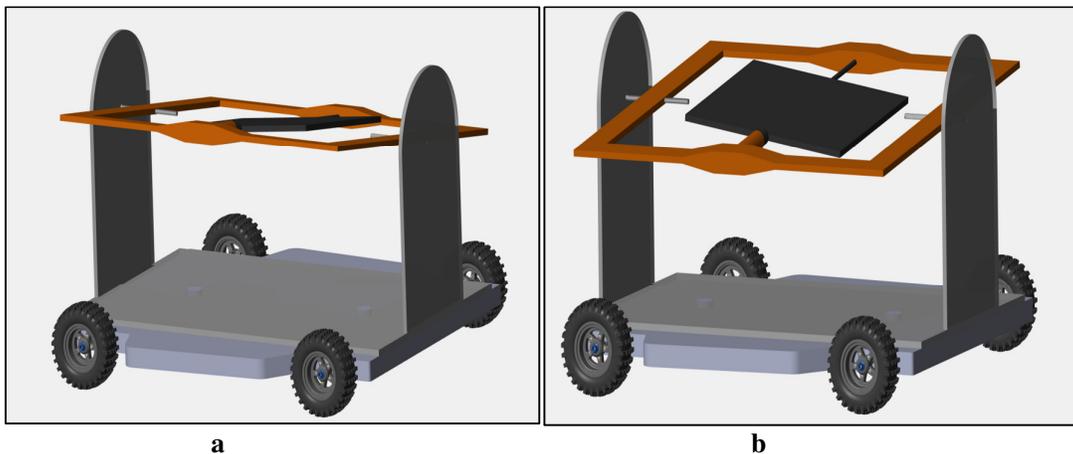
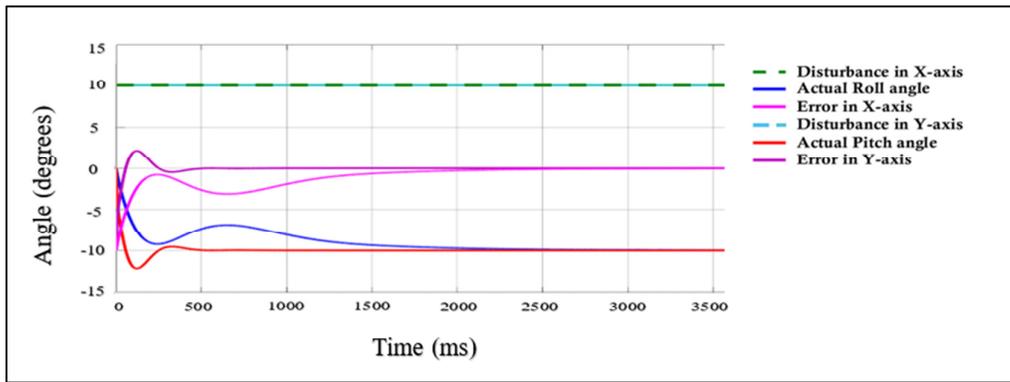


Fig. 15. Self-balancing of the (platform on the mobile car) structure in the virtual simulation a) under tilt angle (10° on positive X-axis and -10° on negative Y-axis) b) under tilt angle (-10° on negative X-axis and 10° on positive Y-axis).

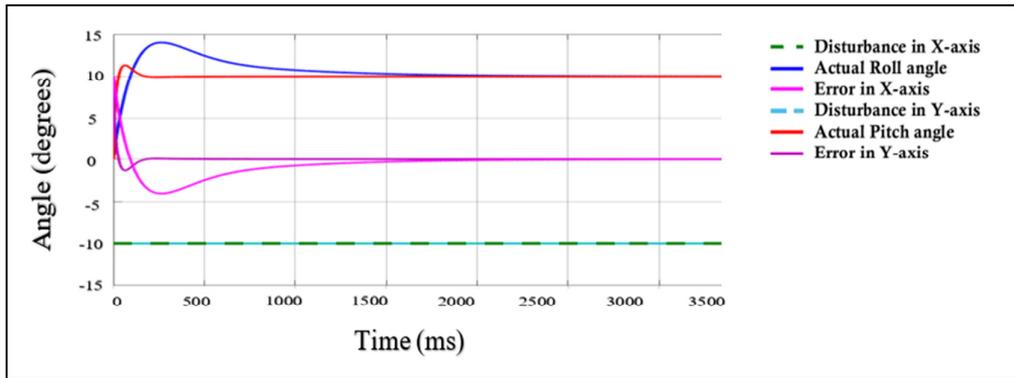
**Case Four: Self-balancing attitude of the simulated model under the influence of road disturbances scenario in the biaxial (similar tilt angles in X-axis and Y-axis).**

The self-balancing behavior of the simulated model for the self-balancing platform on the mobile car as a result of the platform's simulated model subjected to disturbances (tilt angle 10° on positive X-axis and 10° on positive Y-axis) and

(tilt angle -10° on negative X-axis and -10° on negative Y-axis) shown in figure 16. The self-balancing of the (platform on the mobile car) structure in the virtual simulation environment under tilt angle (10° on positive X-axis and 10° on positive Y-axis) and tilt angle (-10° on negative X-axis and -10° on negative Y-axis) is shown in figure 17: (a, b).



A. under tilt angle ( $10^\circ$  on positive X-axis and  $10^\circ$  on positive Y-axis).



B. under tilt angle ( $-10^\circ$  on negative X-axis and  $-10^\circ$  on negative Y-axis).

Fig. 16. Self-balancing behavior of simulated model on (similar tilt angles in X-axis and Y-axis).

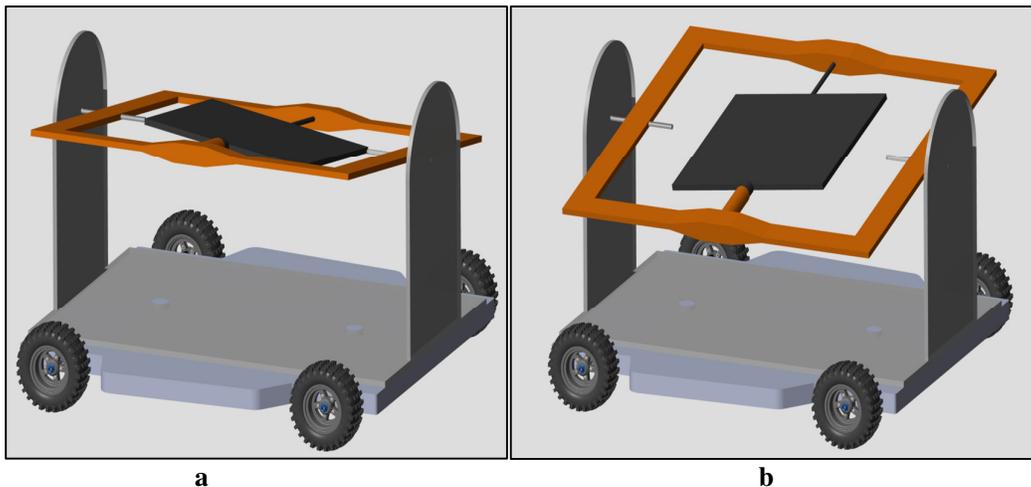


Fig. 16. Self-balancing of the (platform on the mobile car) structure in the virtual simulation a) under tilt angle ( $10^\circ$  on positive X-axis and  $10^\circ$  on positive Y-axis) b) under tilt angle ( $-10^\circ$  on negative X-axis and  $-10^\circ$  on negative Y-axis).

From the previous three cases, simulation results showed that the angle at the beginning of the error curve represents the angle value that the DC servo motor will rotate the respective plate (angle averse to the tilt angle direction from

corresponding disturbances signal but in the same value) in order to overcome the tilt angle that platform exposed to. By comparing the angle in (either actual Roll angle or actual Pitch angle or both actual angles simultaneously) with the angle

in the corresponding error curve, it could be observed that they are equal in value and direction. In other words, the self-balancing of the platform has been achieved. On the other hand, the recovery time (in all cases) that platform takes to set the error to zero, overcome the tilt angle and achieve self-balancing is (less than two milliseconds). Low recovery time (low self-balancing time) is attributable to the PID controllers' efficiency in handling the platform's tilt (X-axis, Y-axis, and biaxial).

Simulation results also indicate to: Firstly, the platform's simulated model overall response under the influence of disturbance scenarios in (X-axis, Y-axis, or biaxial) has done smoothly and without any evident vibration. Secondly, Dispense the complex design of the DC servo motor by SolidWorks and replace it with a simulated model that did not affect the platform's performance. On the contrary, each DC servo motor proved its ability to determining torque to shift the plate's direction (counter the tilt angle direction) with proper synchronization to disturbance occurrence. Thirdly, adding revolute primitive/ sensing properties makes the system more intelligent and effectively measures the platform's angle (actual Roll angle or actual Pitch angle or both actual angles) after self-balancing without adding a sensor. Finally, the self-balancing of the (platform on the mobile car) structure in the virtual simulation represents the visualization of self-balancing in the top plate, middle plate, and both plates, respectively. Key values of simulation results were provided from MATLAB/Scope and shown in Table 2.

For achieving the results above, the PID controller's parameters (for both PID controllers of the self-balancing platform simulated model) are tuned and optimized by the Genetic Algorithm method (GA)[26]. The following parameter values are shown in Table 3 gave the best response in terms of maximum peak overshoot and response time. Genetic Algorithm parameters are illustrated in Table 4. Table 5.5shows a comparison between our self-balancing platform on the mobile car with other article references.

**Table 2,**  
**The key values of simulation results.**

<b>Rise time (in all cases)</b>	114 to 175ms
<b>Overshoot (in all cases)</b>	12% when tilted in one axis, 15% when tilted in two axes.
<b>Settling time</b>	237ms
<b>Steady state error</b>	0.09 to 0.74°

**Table 3,**  
**Optimized PID Controllers' Parameters of simulated model**

	<b>Roll angle</b>	<b>Pitch angle</b>
<b>P</b>	45	32
<b>I</b>	73	59
<b>D</b>	0.58	0.54

**Table 4,**  
**Genetic Algorithm parameters.**

<b>S.No.</b>	<b>GA Parameters</b>	<b>Value/ Method</b>
1.	Optimization tool	Matlab Genetic Algorithm Toolbox
2.	Variable boundaries [Kp Ki Kd]	For Roll angle PID controller Lower [0,0,0,], Upper [ 150 ,350, 1] For Pitch angle PID controller Lower [0,0,0,], Upper [ 100 ,200, 1] ❖ Lower and upper boundaries values have been taken from primary PID parameters(for both PID controllers) tuned by trial and error.
3.	Maximum number of Population	60 [26].
4.	Size generations	100 [26].
5.	Fitness value	Cost function/ Integral of time multiplied by absolute error (ITAE).
6.	Selection method	Normalized Geometric Selection[26].
7.	Crossover Method	Arithmetic Crossover [26].
8.	Mutation method	Uniform Mutation [26].

**Table 5,**  
**Comparison between our self-balancing platform on the mobile car with other article references.**

	Disturbance type	Self-balancing time
<b>Our simulated model</b>	Disturbances signal as tilt angle from (-10 and 10) degrees on (X-axis or Y-axis or biaxial).	less than 2 milliseconds
[9]	The disturbance that was used is the positive angle value (5 degrees) of the two axes.	0.7 seconds
[10]	Pump =3 and 5 mm	2.6, 1.6, 1.2 seconds in (X- axis, Y-axis, Z-axis), respectively
[11]	The platform's design is 6DOF, but the platform was only tested when the base was rotated 26 degrees with the X-axis and the Y-axis.	50 seconds
[12]	tilt angle= 2-25 degrees in (X-axis & Y-axis).	5 seconds in both axes

## 5. Conclusion

In this paper, a simulated model of self-balancing on the mobile car has been suggested and designed. The work results showed that when the platform is exposed to the disturbance in (X-axis, Y-axis, or biaxial), the controller's decision (angle value and direction of the servo motor) depending on error measurements to overcome the platform's unbalanced situation and reduce recovery time. Moreover, Optimization of the PID controllers' coefficients in both (simulated model) by the genetic algorithm method effectively affected the performance of the platform, as the platform system is stable and the platform was able to compensate for the tilt angle in (X-axis, Y-axis, and biaxial) and overcome the error in a time that does not exceed two milliseconds.

## 6. Future Work

Future work includes changing the disturbance profile of (X-axis, Y-axis, or biaxial) into diverse values of tilt angles (between each angle and another a period of time) on the positive and negative side of the axis. Also, add new elements to the platform's dynamic modelling design to study the self-balancing performance in six DOF.

## 7. References

- [1] M. M. Nadeem, S. U. Khan, and D. Mazhar, "Dynamic Self Stabilizing Mobile Platform," no. Ieec, 2019.
- [2] V. Popelka, "A self stabilizing platform," Proc. 2014 15th Int. Carpathian Control Conf. ICC 2014, pp. 458–462, 2014, doi: 10.1109/CarpathianCC.2014.6843648.
- [3] Y. Liu, Z. Q. Zhu, and D. Howe, "Instantaneous torque estimation in sensorless direct-torque-controlled brushless DC motors," IEEE Trans. Ind. Appl., vol. 42, no. 5, pp. 1275–1283, 2006, doi: 10.1109/TIA.2006.880854.
- [4] Mohapatra, S., Srivastava, R., & Khera, R. (2019). "Implementation of a two-wheel self-balanced robot using MATLAB simscape multibody". Second International Conference on Advanced Computational and Communication Paradigms (ICACCP)1–3. DOI:10.1109/ICACCP.2019.8883007.
- [5] Walica, P. N. D. (2020). "Design and Realisation of the Simulation Model of the Stewart Platform using the MATLAB-Simulink and the Simscape Multibody Library". 2020 21th International Carpathian Control Conference (ICCC).
- [6] Haidar, A. M. A., Benachaiba, C., & Zahir, M. (2013). "Software interfacing of servo motor with microcontroller". Journal of Electrical Systems, 9(1), 84–99.
- [7] A. D. Assi, "Study of Transverse and Longitudinal Crack Propagation in Human Bone Using the Finite Element Method with MATLAB Abdullah Dhayea Assi," vol. 15, no. 2, pp. 44–49, 2019.
- [8] Alkamachi, A. (2020). "Integrated SolidWorks and Simscape platform for the design and control of an inverted pendulum system". Journal of Electrical Engineering, 71(2), 122–126. DOI:10.2478/jee-2020-0018.Q
- [9] M. B. Mat Ali, "Development of Self Balancing Platform," 2013.
- [10] G. Madhumitha, R. Srividhya, J. Johnson, and D. Annamalai, "Physical modeling and control of self-balancing platform on a cart," Proc. 2016 Int. Conf. Robot. Curr. Trends

- Futur. Challenges, 2017, doi: 10.1109/RCTFC.2016.7893410.
- [11] G. Adel, A. Iop, C. Ser, M. Sci, and G. A. Aziz, "Stability Control Investigation of a Self-Balancing Platform on the Robot Smart Car Using Navigation Parameters Stability Control Investigation of a Self-Balancing Platform on the Robot Smart Car Using Navigation Parameters," 2020, doi: 10.1088/1757-899X/765/1/012066.
- [12] M. Alkhedher, U. Ali, and O. Mohamad, "Modeling, simulation and design of adaptive 6DOF vehicle stabilizer," 2019 8th Int. Conf. Model. Simul. Appl. Optim. ICMSAO 2019, pp. 6–9, 2019, doi: 10.1109/ICMSAO.2019.8880417.
- [13] R. Chen, Y. Ou, W. Fang, Y. Shi, and L. Liu, "Simulation analysis of a self-balancing hydraulic platform for agricultural machinery in mountainous regions," J. Eur. des Syst. Autom., vol. 53, no. 2, pp. 203–211, 2020, doi: 10.18280/jesa.530206.
- [14] Thomas, N., & Poongodi, P. (2019). "Position Control of DC Motor Using Genetic Algorithm Based PID Controller". Proceedings of the World Congress on Engineering 2009 Vol II.
- [15] Mirzal, A., Yoshii, S., & Furukawa, M. (2012). "PID Parameters Optimization by Using Genetic Algorithm". International Journal of Emerging Technology and Advanced Engineering. <http://arxiv.org/abs/1204.0885>.
- [16] Mantri, G. & Kulkarni, N. R. (2013). "Design and Optimization of Pid Controller Using Genetic Algorithm". International Journal of Research in Engineering and Technology, 02(06), 926–930. DOI:10.15623/ijret.2013.0206002.
- [17] R. A. Kadhim, "Design and Simulation of Closed Loop Proportional Integral (PI) Controlled Boost Converter and 3-phase Inverter for Photovoltaic (PV) Applications," Al-Khwarizmi Eng. J., vol. 15, no. 1, pp. 10–22, 2019, doi: 10.22153/kej.2019.06.001.
- [18] K. Liu, M. Bai, and Y. Ni, "Two-wheel self-balanced car based on Kalman filtering and PID algorithm," 2011 IEEE 18th Int. Conf. Ind. Eng. Eng. Manag. IE EM 2011, no. PART 1, pp. 281–285, 2011, doi: 10.1109/IEEM.2011.6035158.
- [19] Sabah, A. "Design of Nonlinear PID Neural Controller for the Speed Control of a Permanent Magnet DC Motor Model based on Optimization Algorithm". Al-Khwarizmi Engineering Journal, Vol. 10, No. 1, P.P. 72-82 (2014).
- [20] Abd-Alkarim and Shereen F., "Application of fuzzy logic in servo motor.," Al-Khwarizmi Engineering Journal vol. 3, no. 2, pp. 8-16, 2007.
- [21] Ayasun, S., & Karbeyaz, G. (2008). " DC motor speed control methods using MATLAB/Simulink and their integration into undergraduate electric machinery courses". Journal of Computer Applications in Engineering Education, 15(4), 347–354. DOI:10.1002/cae.20151.
- [22] M. A. Rashidifar, A. A. Rashidifar, and D. Ahmadi, "Modeling and Control of 5DOF Robot Arm Using Fuzzy Logic Supervisory Control," IAES Int. J. Robot. Autom., vol. 2, no. 2, pp. 56–68, 2013, doi: 10.11591/ijra.v2i2.2974.
- [23] Munadi and M. A. Akbar, "Simulation of fuzzy logic control for DC servo motor using Arduino based on MATLAB/Simulink," Proc. 2014 Int. Conf. Intell. Auton. Agents, Networks Syst. Ina. 2014, pp. 42–46, 2015, doi: 10.1109/INAGENTSYS.2014.7005723.
- [24] Giacotto, A. (2019). "MatlabSimscape Multibody model based simulation for Mechatronic systems". M.Sc. Thesis, Mechatronics Engineering, Polytechnic University of Turin. Italy.
- [25] A. N. Barakat, K. A. Gouda, and K. A. Bozed, "Kinematics analysis and simulation of a robotic arm using MATLAB," Al-Khwarizmi Eng. J., vol. 15, no. 1, pp. 10–22, 2017, doi: 10.1109/CEIT.2016.7929032.
- [26] Meena, D., K. & Chahar, S. (2017). "Speed Control of DC Servo Motor Using Genetic Algorithm". IEEE International Conference on Information, Communication, Instrumentation and Control (ICICIC).DOI:10.1109/ICOMICON.2017.8279122.

## محاكاة منصة ذاتية التوازن على سيارة المتحركة

بشرى عامر توفيق \* ماهر يحيى سلوم\*\*

احمد محروس\*\*\*

\* قسم هندسة الميكاترونكس/ كلية هندسة الخوارزمي/ جامعة بغداد

\* البريد الالكتروني: [bushra.a.tawfeeq@gmail.com](mailto:bushra.a.tawfeeq@gmail.com)\*\* البريد الالكتروني: [drmahir@kecbu.uobaghdad.edu.iq](mailto:drmahir@kecbu.uobaghdad.edu.iq)\*\*\* البريد الالكتروني: [ahmed78@kecbu.uobaghdad.edu.iq](mailto:ahmed78@kecbu.uobaghdad.edu.iq)

## الخلاصة

في السنوات الأخيرة، أصبحت منصة التوازن الذاتي واحدة من أكثر المرشحين شيوعاً لاستخدامها في العديد من التطبيقات مثل الطيران والمجالات الطبية الحيوية والصناعة وما إلى ذلك. في هذا البحث تم تقديم موقف التسوية الذاتية في (المحور  $X$  أو المحور  $Y$  أو كلا المحورين) تحت تأثير اضطراب الطريق. لمحاكاة أداء النظام الأساسي المتوازن ذاتياً أثناء الإمالة، تم استخدام تكامل بين أدوات *Solidworks* و *Simscape* و *Simulink* في *MATLAB* تم رسم النموذج الديناميكي للنظام الأساسي في *SolidWorks* وتم تصديره كملف *STEP* مستخدم في بيئة *Simscape Multibody*. يتم التحكم في النظام باستخدام وحدة التحكم في المشتقات النسبية المتكاملة (*PID*) للحفاظ على مستوى المنصة وتعويض أي اضطرابات على الطريق. تم تصميم العديد من سيناريوهات اضطرابات الطريق في المحور السيني أو المحور الصادي أو كلا المحورين (زوايا الانحدار والدوران) لفحص فعالية وحدة التحكم. تشير نتائج المحاكاة إلى أن المنصة أكملت التسوية الذاتية تحت تأثير الاضطراب (١٠ درجة و -١٠ درجة) على المحور السيني والمحور الصادي وكلا المحورين في أقل من ٢ ملي ثانية. لذلك، فإن نموذج المحاكاة لمنصة التوازن الذاتي المقترح يتمتع بدقة تسوية عالية ويلبي المتطلبات التشغيلية على الرغم من تصميمه البسيط.