A Model of 3-DOF Stewart Auto-Balancing System for Low-Cost and High Reliable Embedded Applications

Anh, Nguyen Pham The
Faculty of Computer Engineering
VNUHCM - University of Information Technology
Ho Chi Minh City, Vietnam
18520460@gm.uit.edu.vn

Hoang, Bui Huy
Faculty of Computer Engineering
VNUHCM - University of Information Technology
Ho Chi Minh City, Vietnam
18520773@gm.uit.edu.vn

Hoai, Nguyen Thi
Faculty of Information Science and Engineering
VNUHCM - University of Information Technology
Ho Chi Minh City, Vietnam
hoaint@uit.edu.vn

Duy, Doan
Faculty of Computer Engineering
VNUHCM - University of Information Technology
Ho Chi Minh City, Vietnam
duyd@uit.edu.vn

***Abstract*—Recently, the automotive industry has increased rapidly and provided a number of efficient solutions for daily life, science, and manufacture. An auto-balancing system is such an automated solution that attracts much attentions from scientists and engineers. Designing a 3-dimension auto-balancing system is challenging in terms of accuracy, responsiveness, and noise reduction. In this paper, we introduce a model of a three-dimension auto-balancing system with high accuracy and low noise. We applies the inverse Kinematics for high accuracy of motor control and sensor fusion technique to reduce noise and increase reliability with sensor fusion of output data from sensors. The proposed system is built and simulated using Matlab model. Simulation results show that the proposed system can significantly reduce noise of sensors and achieve a response of fewer than two milliseconds.**

***Index Terms*—Three-Dimension balance, Auto-Balancing, PID, Smart Balancing, IMU, sensor fusion.**

I. Introduction

In recent decades, automotive industry has developed rapidly and automated devices have become popular in daily life and science. Automated doors in houses, automated pumping system at water companies or automated car assembling systems are typically examples. Auto-balancing system is also an important design that is ubiquitously implemented in different applications. Complex complex auto-balancing systems are integrated into cars, air planes, or industrial lifters to increase safety and stability.

Designing motion platform systems was attractive to researchers and engineers early from 1965 when Stewart D. introduced an 6-DOF (degree of freedom) platform controlled by combination of six motors for flight simulation [1]. However, this platform are unaffordable in most of applications as the cost of motion platform barely depends on the number of actuators. In 1996, Yang et al. proposed an 3-DOF motion platform [2] for low-cost applications. The 3-DOF model, which requires only three actuators comparing with six ones in the 6-DOF model, effectively shows high performance and significant cost reduction.

Stewart platform is widely applied in motion simulations with both models of 6-DOF and 3 DOF. Stewart model is also potential for implementing of auto-balancing systems. In 2018, Kittipong Yaovaja used Stewart Platform in the implementation of ball balancing [3] using image processing and Fuzzy supervisor of PID for controlling. Since image processing is faced with several obstacles of environment, the accuracy and responsiveness are not high.

In recent years, auto-balancing systems have been improved the accuracy with new techniques of PID model, machine learning algorithms, and precise mechanism structures. Two-wheel self-balancing robots are implemented toward applications for robotics and disability-support devices [4]–[6], which applies PID for motor control. New researches on auto-balancing employ machine learning techniques to improve the reliability [7]. Systems with machine learning require complex hardware and software platforms for implementation, which is not suitable for low-cost applications.

Contribution We are targeted to provide a model of designing a 3D auto-balancing system with high accuracy and fast responsiveness for low-cost applications. Challenging problems of 3D-auto balancing system are estimating the angles of the balancing platform, reducing the base noise of input data from sensors, and controlling the accurate rotation of motors. Dealing with these problems, we suggest a model of 3-DOF Stewart platform with sensor fusion algorithm for reducing noise of IMU sensor and enhanced PID model for motor controlling. The proposed balancing model is suitable for not only industrial fields, but the low-cost devices such as

safe chair for drivers and passengers, gurneys in hospital.

The rest of the paper is organized as follows. Section II describes the overview of the proposed system and the simulated system on Matlab. Section III explores filtering techniques to reduce base noise of sensor. Section IV provides the description of Stewart platform and the model to calibrate the accuracy of the 3D auto-balancing system. Then, 3-PID model for motor controlling is described in Section V. Finally, in Section VI is the conclusion and discussion for future work.

II. Proposed System Model

The proposed auto-balancing system has two stages: (1) Sensor data processing and (2) Motor controlling, as depicted in Fig. 1.

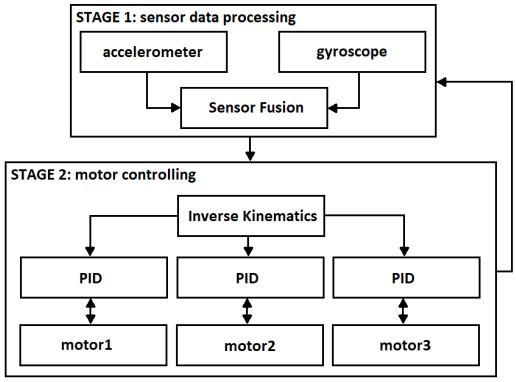

Fig. 1. Overview of the control system

The first stage consists of sensors attached to the upper platform in order to measure the balance situation. Sensors are to estimate the pitch and roll angles of the platform that indicate the pitch and roll movements. MPU-6050 [8] sensors, which are integrated with three-axis of accelerometers and gyroscopes, are used via I2C protocol. One of the challenges of this system is the effect from the motors' rotation to the sensors; namely, rotating motors generate vibration as noise to the accelerometers. Therefore, a sensor fusion is implemented to reduce sensor's noise and gyroscope drift. Processed data from the first stage allows to increase the control accuracy and speed at the second stage. Sensor fusion techniques are going to detail in Section III.

At the second stage, inverse Kinematics is employed as the general control for motors. Input data of roll and pitch angles is used to calculate θ_{pitch} and θ_{roll} as in Equation 1 and 2. Such equations are guaranteed to maintain the platform balanced.

$$\theta_{pitch} = init_{pitch} - MPU_{pitch} \tag{1}$$

$$\theta_{roll} = init_{roll} - MPU_{roll} \tag{2}$$

where MPU_{pitch} and MPU_{roll} are values of pitch and roll angles estimated from the first stage, $init_{roll}$ and $init_{pitch}$ are the initial balanced angles of the platform.

Based on the calculated θ angles, The inverse Kinematics is applied to set the target set-points for the three PID controllers. The operation of inverse Kinematics is going to detail in Section IV. Set-points represent the situation when the platform is balanced. PID controllers then simultaneously drive three motors to reach set-points, that is, setting the platform balanced. The controlling model of PID is shown in Fig. 2.

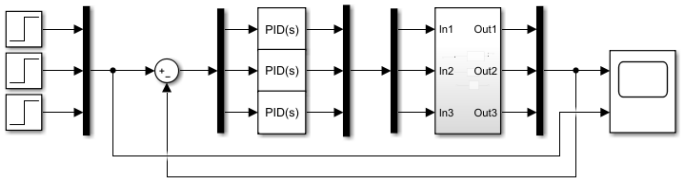

Fig. 2. PID control scheme

After all motors reach their set-point, the system returns to a closed loop to update the new θ angles to continuously maintain the balance of the platform.

III. SENSOR FUSION

As mentioned above, MPU-6050, a low-cost inertial measurement unit (IMU) consisting of sensors and sensor fusion software, is used to detect balance situation of the platform. Since it is integrated with accelerometers and gyroscopes, this unit is sensitive with vibration noise from motors rotation. To reduce such kind of noise for high accuracy of control, we employed the Kalman filter in comparison with the Complementary filter.

A. Problem of vibration noise

Using 3-axis accelerometer (each axis of accelerometer represents an end point of the platform), we can estimate pitch and roll angles using Equations 3 and 4.

$$acc_{pitch} = atan * (\frac{-acc_x}{\sqrt{(acc_y)^2 + (acc_x)^2}}) \tag{3}$$

$$acc_{roll} = asin(\frac{acc_y}{acc_z}) \tag{4}$$

where acc_x, acc_y, and acc_z are gravitational acceleration values on 3-axes (x, y, z), respectively.

There exists another way to obtain these values of pitch and roll angles using angular velocity from gyroscopes. Equations 5 and 6 show the calculation of these values using x and y axes of gyroscope.

$$gyro_{pitch}(t) = \int_0^t gyro_y(t)dt + bias_{pitch} \tag{5}$$

$$gyro_{roll}(t) = \int_0^t gyro_x(t)dt + bias_{roll} \tag{6}$$

where $gyro_y$ and $gyro_x$ are gyroscope values at the time t on x and y axes, respectively; gyroscope data has units of angular velocity and $bias_{roll}$, $bias_{pitch}$ are gyroscope bias.

However, the disadvantage of calculations in Equations (3) and (4) is that accelerometor is vulnerable to noise from motor vibration causing inexact estimation of accelerometer while Equations (5) and (6) are faced with the phenomenon that gyroscope outputs tend to be drifted over time and not constant even if gyroscopes stay stable (see Fig. 3). Since the reliability of the estimated values of pitch and roll angles is very important to control motors accurately, we need a solution called IMU sensor fusion to eliminate such disadvantages of accelerometor and gyroscope. Complementary and Kalman filters [9] are often candidate to implement the IMU sensor fusion.

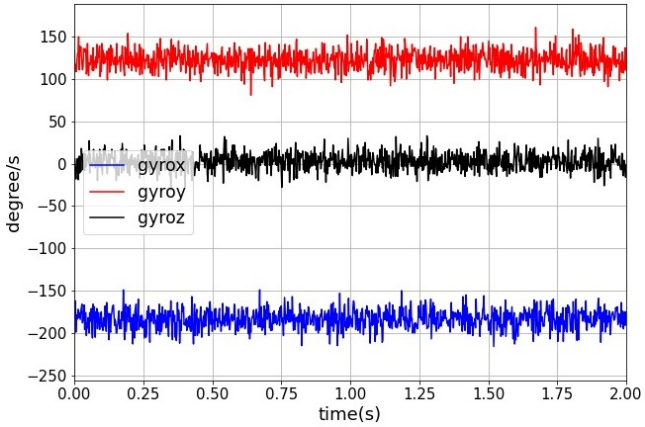

Fig. 3. Gyroscope's noise real-time test

B. Complementary Filter

Complementary Filter uses values of gyroscope as the major factors to minimise the oscillation (motor vibration) at the output, and offset a small factor of accelerometor to compensate the drift of gyroscope as showed in Equations 7 and 8.

$$roll = \alpha(gyro_{roll}) + (1-\alpha)acc_{roll} \tag{7}$$

$$pitch = \alpha(gyro_{pitch}) + (1-\alpha)acc_{pitch} \tag{8}$$

where α is the filter coefficient ($0 < \alpha < 1$). When α gets closer to 1, the filter shows better filtering performance, but more drift.

C. Kalman Filter

The true state (x_k) and its corresponding measurement (z_k) at time k of Kalman filter are defined in Equations 9 and 10.

$$x_k = F_k x_{k-1} + Bu_k + w_k \tag{9}$$

$$z_k = Hx_k + v_k \tag{10}$$

where F_k is the state transition model, B_k is the control-input model, w_k is process noise, H_k is the observation model, and v_k is observation noise.

In case of 6-DOF IMU [10] (three values from accelerometer and three values from gyroscopes), Equation 9 and 10 can be re-written as in Equation 11 and 12, where output θ and bias $\dot{\theta}_b$ at time k are based on the accelerometer and gyroscope, and the bias is the amount that the gyroscope has drifted.

$$\begin{pmatrix}\theta \\ \dot{\theta}_b\end{pmatrix}_k = \begin{pmatrix}1 & -\Delta t \\ 0 & 1\end{pmatrix}\begin{pmatrix}\theta \\ \dot{\theta}_b\end{pmatrix}_{k-1} + \begin{pmatrix}\Delta t \\ 0\end{pmatrix}\dot{\theta}_k + w_k \tag{11}$$

$$z_k = \begin{pmatrix}1 & 0\end{pmatrix}\begin{pmatrix}\theta \\ \dot{\theta}_b\end{pmatrix}_k + v_k \tag{12}$$

D. Evaluating simulation

For comparison, we implement a complementary filter with coefficient α equal to 0.98 and a Kalman filter with parameters (Q_{angle}, Q_{bias}, R) set at 0.001, 0.003 and 0.03, respectively. The loop time of both filters is set to $0.002s$. These constants are chosen so that the filtering efficiency is acceptable.

Fig. 4 and 5 show the testing results on the output drift of each filter. Accordingly, the drift amount of the complementary filer is approximately 0.15 (degree) in a period of 3000 seconds. This is a little higher than Kalman filter which results with 0.13 (degree) of drift amount. Actually, there is not much different in terms of drifting.

Furthermore, comparing to non-filter version calculated by Equations 3 and 4, both complementary and Kalman filters perform significant improvement in isolating the accelerometor noise as showed in Fig.6 and 7. The result is also maintained in case of noise addition by motors vibration (see Fig. 8 and 9). This indicates the high reliability of the angle estimation.

Additionally, the complementary take advantage of faster react-time to the input changes while Kalman filter gives higher stability and accuracy on the output result. Therefore, the Kalman filter is more preferable for the implementation of this auto balancing system.

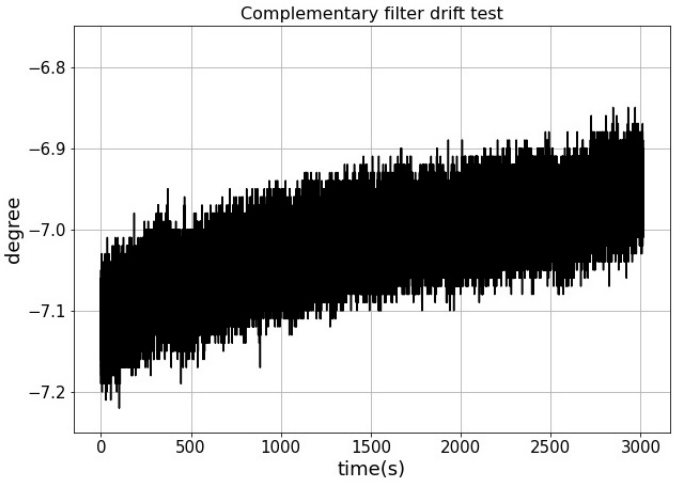

Fig. 4. Complementary Filter Drift Test

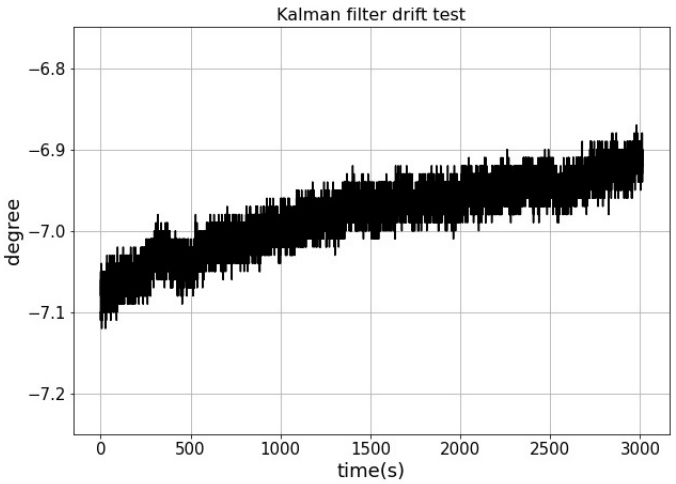

Fig. 5. Kalman Filter Drift Test

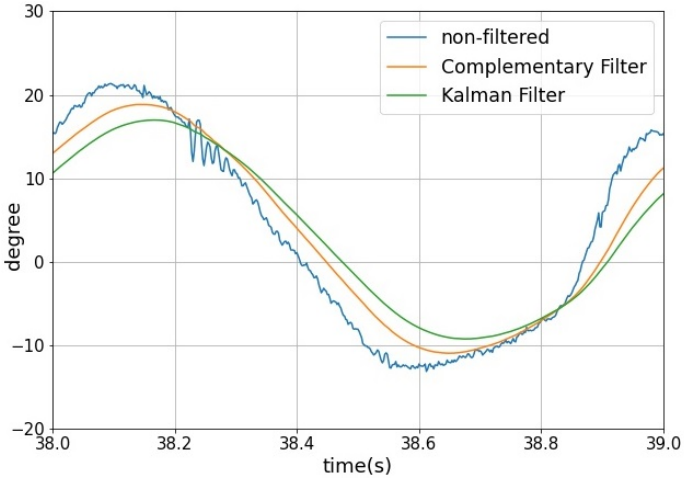

Fig. 7. Roll Dynamic Test

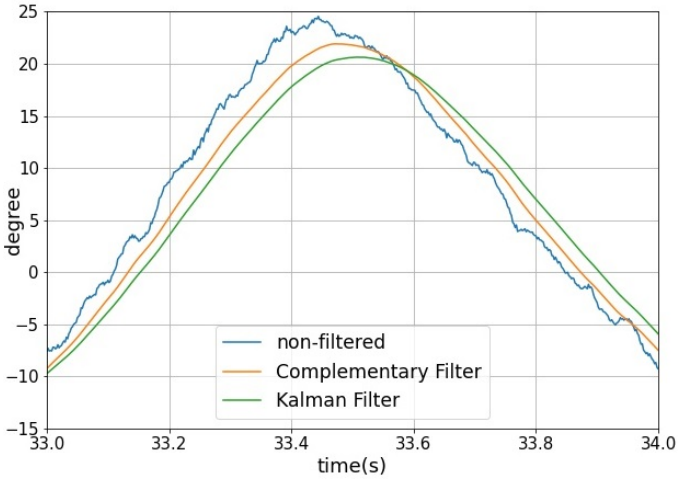

Fig. 6. Pitch Dynamic Test

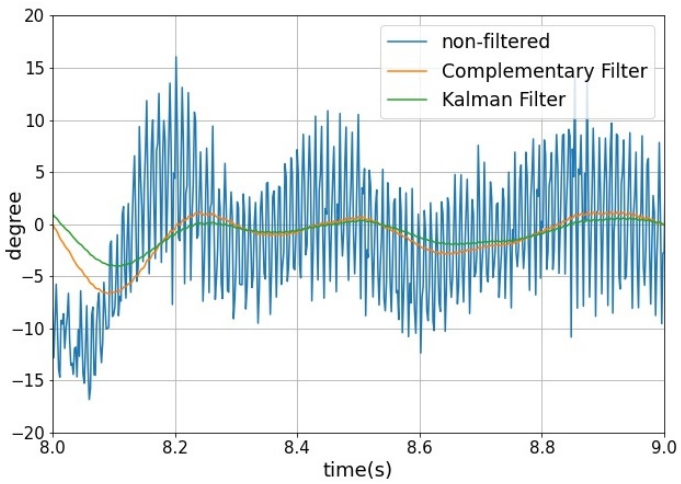

Fig. 8. Pitch Dynamic Test with noise

IV. 3-DOF Stewart Parallel Manipulator

This section is to describe in detail the theory of the 3D auto-balancing system using inverse Kinematics and Stewart platform model.

A. Inverse Kinematics

Kinematics is widely applied in problems that need to be solved about coordinate systems such as Robotics, Automation, etc. As in this work, we are dealing with designing a 3-D auto-balancing system, a popular problem of coordinate system. Specifically, we apply Kinematics theory to the processing of Stewart platform data to give the angular parameter values in order to control the motors of the system. The 3-D auto-balancing system is implemented with three motors, which rapidly and intensively rotate to maintain a balancing platform. To achieve this system, we need to guarantee that the Stewart platform coordinate data quickly returns the angular parameter value of each motor, and then the exact rotation angle of each motor is obtained.

Inverse Kinematics [12] is one of the possible solutions for the problem of extracting the exact value of rotation angle. Fig. 10 shows the direct view of a traditional inverse Kinematics. According to the inverse Kinematics theory, angles θ_1 and θ_2 is properly calculated when the endpoint coordinates $C1$, $C2$ and $C3$ are determined.

In this research, we are targeted to determine the rotation angle of motors, and therefore angle θ_1 is took into account only. Applying the inverse Kinematics theory, our 3-D auto-balancing system can be modeled as showed in Fig. 11 where $A1$, $A2$, and $A3$ represent to the center of the three motors.

In Fig. 11, the balance of the platform formed by $C1$, $C2$, and $C3$ depends on the rotation angle values of θ_1, θ_2, θ_3 at $A1$, $A2$, and $A3$, respectively. At the point coordinates $C1$, $C2$, and $C3$, we can apply the Inverse Kinematics to find out the values of θ_1, θ_2, θ_3 that keep the platform balanced [10]. These values of rotation angle are then used to control the three motors for balancing.

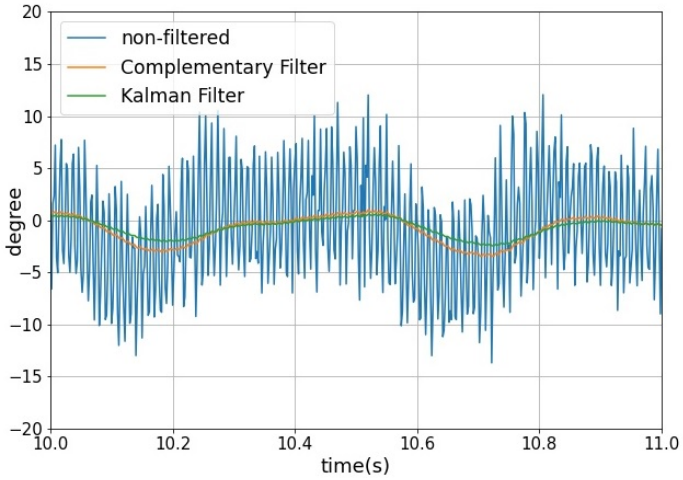

Fig. 9. Roll Dynamic Test with noise

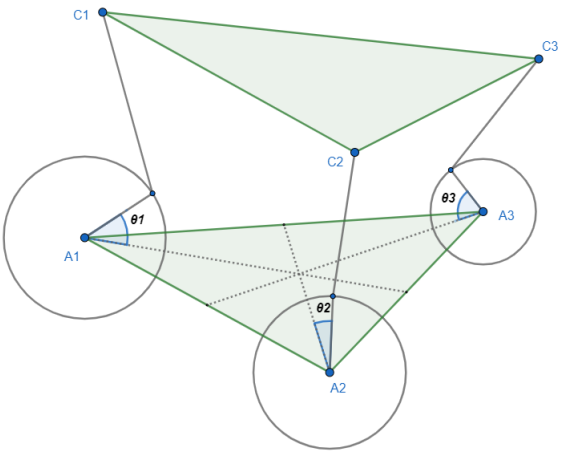

Fig. 11. Project overview based on building virtual models

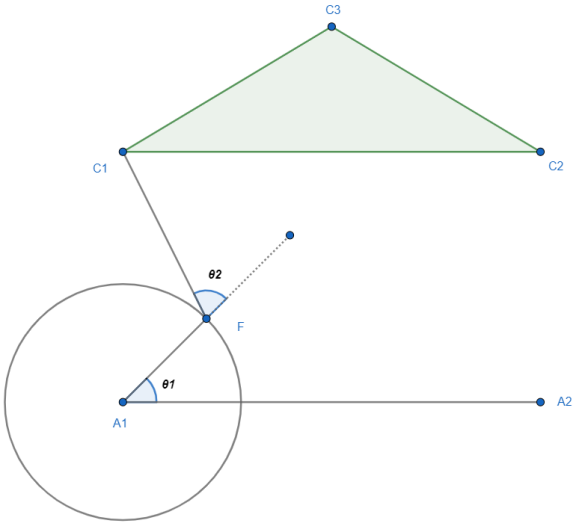

Fig. 10. Direct view project example

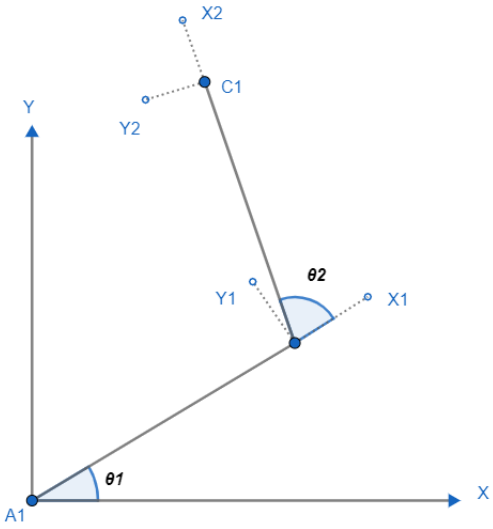

Fig. 12. Axis model of 1-DOF

B. Geometric algebra

In the problem of 3-D auto-balancing, each motor has to rotate in coordination with the other motors. In order to achieve a accurate calculation of these rotations, mathematical model of the three motors is built using geometric algebra. Fig. 12 depicts the geometric model (1-DOF axis) of one motor in which θ_1 is the rotation angle at the center of the motor. After that, the premise for axis 2 and axis 3 (of the other motors) will be calculated similarly.

In this work, the interesting value is θ_1 in Fig. 12 which is used for calculating the set-point by Kinematics. When modeling a kinematic sequence and specifically identifying the frames to be attached to each body, it is common to use the Denavit-Hartenberg (DH) parameters [14], [15].

Based on the DH table,we can set 4x4 matrices as follows:

$$ {}^{0}T_1 = \begin{bmatrix} cos\theta_1 & -sin\theta_1 & 0 & l_1 cos\theta_1 \\ sin\theta_1 & cos\theta_1 & 0 & l_1 sin\theta_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{13} $$

$$ {}^{1}T_2 = \begin{bmatrix} cos\theta_2 & -sin\theta_2 & 0 & l_2 cos\theta_2 \\ sin\theta_2 & cos\theta_2 & 0 & l_2 sin\theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{14} $$

$$ {}^{0}T_2 = {}^{0}T_1 * {}^{1}T_2 \tag{15} $$

TABLE I
DENAVIT – HARTENBERG PARAMENTERS

Frame No.	a_i	α_i	d_i	θ
1	l_1	0	0	θ_1
2	l_2	0	0	θ_2

$$ {}^0T_2 = \begin{bmatrix} cos(\theta_1{+}\theta_2) & -sin(\theta_1{+}\theta_2) & 0 & l_1cos\theta_1{+}l_2cos(\theta_1{+}\theta_2) \\ sin(\theta_1{+}\theta_2) & cos(\theta_1{+}\theta_2) & 0 & l_1sin\theta_1{+}l_2sin(\theta_1{+}\theta_2) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{16} $$

Equation 16 allows us to determine the coordinate values of the set-point using Equation 17 and its transformation in Equation 18 as follows:

$$ \begin{bmatrix} X \\ Y \end{bmatrix} = \begin{bmatrix} l_1cos\theta_1 + l_2cos(\theta_1{+}\theta_2) \\ l_1sin\theta_1 + l_2sin(\theta_1{+}\theta_2) \end{bmatrix} \tag{17} $$

$$ X^2 + Y^2 = l_1^2 + l_2^2 + 2l_1l_2cos\theta_2 \tag{18} $$

Equation 18 is applied to inverse Kinematics to find out the rotation angles θ_1 and θ_2 for motor control as follows:

$$ cos\theta_2 = \frac{X^2 + Y^2 - l_1^2 - l_2^2}{2l_1l_2} \tag{19} $$

$$ \theta_2 = cos^{-1} * (\frac{X^2 + Y^2 - l_1^2 - l_2^2}{2l_1l_2}) \tag{20} $$

$$ \theta_2 = \pm 2atan2 * \sqrt{\frac{(l_1 + l_2^2)^2 - (X^2 + Y^2)}{(X^2 + Y^2) - (l_1 - l_2)^2}} \tag{21} $$

$$ \theta_1 = atan2 * (\frac{Y}{X}) - atan2 * (\frac{l_2 sin\theta 2}{l_1 + l_2 cos\theta_2}) \tag{22} $$

Looking at Fig. 11, Equations 21 and 22 provide a calculation for the rotation angle of *Motor 1* (at A1) based on the point coordinates of C1. Similarly, following the same calculation sequences for the point coordinates of C2 and C3, we can obtain the rotation angles $\theta 2$ and $\theta 3$ for *Motor 2* and *Motor 3* at A2 and A3, respectively [12].

In implementation, actually after calculating θ_1 and θ_2 based on Equations 22 and 21, we change the X and Y coordinates to the rotation angle θ_{roll}, and θ_{pitch} to fit the MPU6050 structure. Since these equations are linear, θ_{roll}, and θ_{pitch} can be obtained easily.

V. 3-PID MODEL FOR MOTOR CONTROL

In this work, PID control is an important part deciding the accuracy in motor control. In fact, PID is commonly applied in automotive control systems to regulate motor outputs, limit oscillation, and suppress overshoot.

The PID formula for the operating value [16] is defined as Equation 23 with K_P is the proportional gain, K_I the integral gain, K_D the derivative gain.

$$ u(t) = k_P e(t) + k_I \int_0^t e(t)dt + k_D \frac{d}{dt} e(t) \tag{23} $$

The three functionalities are highlighted below:

- Proportional adjustment (P): This allows to generate an adjustment signal proportional to the input source error.
- Integral adjustment (I): The proportional adjustment method leaves a very large offset to the output controlling signal. To overcome this phenomenon, we combine the proportional adjustment with the integral adjustment. Integral tuning is generating the tuning signals with the deviation reduced to zero. Accordingly, the smaller the time, the stronger integral adjustment effect.
- Derivative adjustment (P): When the system time is constant or dead time is large, it will be adjusted according to P or P_I. If the response is too slow, we combine it with derivative adjustment. The derivative adjustment allows to generate a tuning signal proportional to the speed of change of input error. The larger the time is, the stronger derivative adjustment range is.

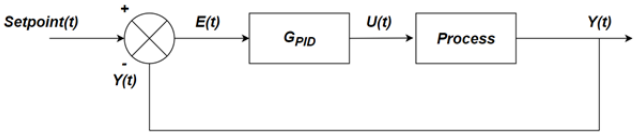

Fig. 13. Closed loop system

Fig. 13 displays the system design of PID control loop for a motor. In this figure, at a time t, input $Setpoint(t)$ is the target value guaranteeing the balance of the platform, which is obtained from inverse Kinematics calculation in Section IV. Combining with Equation 23, We can figure out that if $U(t)$ is a control signal, $E(t)$ is the error value ($E(t) = Setpoint(t) - Y(t)$), $Process$ is operation of the motor to reach the $Setpoint(t)$. The PID controller has a feedback value $Y(t)$.

Fig. 14 shows the response result of the PID motor control on Matlab Simulink. The result indicates a quick response of the control model; namely, the system reach balanced and stable after fewer than two milliseconds .

However in Fig.14 we only simulate with 1 axis of motor to check the stability of the system, the remaining motors have the same mode of operation PID. Actually, our system needs to work with 3 motors at the same time. This means 3 motors have to go through different specialized PID processing. But the returned $Setpoint$ value for each motors are different, because the angle value in Inverse Kinematics calculation mentioned in Section IV is independent.

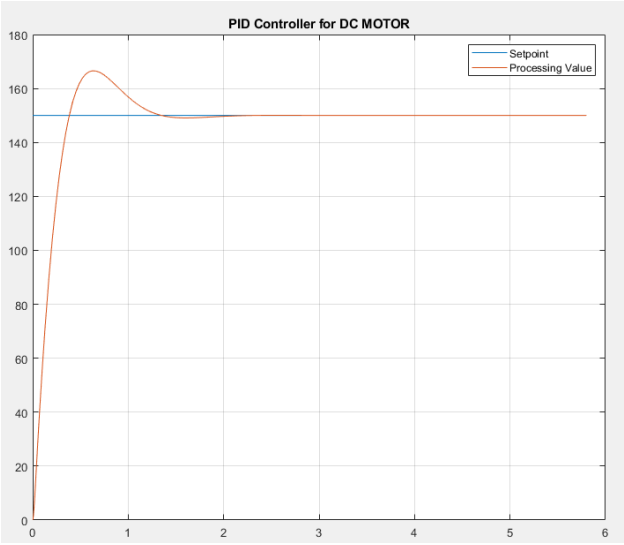

Fig. 14. PID Controller using MATLAB

VI. Conclusion

We have introduced an enhanced design model of 3-DOF auto-balancing system for reliable and low-cost applications. The proposed model applied Kalman filter technique to significantly reduce noise of vibration and inverse Kinematics to achieve high accuracy of PID motor control. Matlab simulations indicate effectiveness of the proposed model with a deep noise reduction and a rapid response of fewer than two milliseconds. In addition, the system is feasible to implemented in low-end embedded systems.

In future work, the system will be practically implemented to get practical calibrations for filter and Kinematics calculation. Especially, the practical implementation is needed to verify the effect of object's load and movement on the balance system. Furthermore, from the practical implementation, we can get data for auto-calibration with more complex technique such as machine learning toward high-class applications.

References

[1] Stewart, D. (1966), "A Platform with Six Degrees of Freedom: A new form of mechanical linkage which enables a platform to move simultaneously in all six degrees of freedom developed by Elliott-Automation", Aircraft Engineering and Aerospace Technology, Vol. 38 No. 4, pp. 30-35.

[2] Yang PH., Waldron K.J., Orin D.E. (1996) Kinematics of a Three Degree-Of-Freedom Motion Platform for a Low-Cost Driving Simulator. In: Lenarčič J., Parenti-Castelli V. (eds) Recent Advances in Robot Kinematics. Springer, Dordrecht.

[3] Kittipong Yaovaja (2018), "Ball Balancing on a Stewart Platform using Fuzzy Supervisory PID Visual Servo Control," Krabi, Thailand.

[4] Souza M.R.S.B., Murofushi R.H., Tavares J.JP.Z..S., Ribeiro J.F., "Comparison Among Experimental PID Auto Tuning Methods for a Self-balancing Robot," Communications in Computer and Information Science, Vol 619, Springer, Cham, 2016.

[5] C. Iwendi, M. A. Alqarni, J. H. Anajemba, A. S. Alfakeeh, Z. Zhang and A. K. Bashir, "Robust Navigational Control of a Two-Wheeled Self-Balancing Robot in a Sensed Environment," in IEEE Access, vol. 7, pp. 82337-82348, 2019.

[6] M. Stănese, M. Şuşcă, V. Mihaly and I. Naşcu, "Design and Control of a Self-Balancing Robot," 2020 IEEE International Conference on Automation, Quality and Testing, Robotics (AQTR), 2020.

[7] Chen Fang, Lirong Cui, "eliability evaluation for balanced systems with auto-balancing mechanisms," Reliability Engineering System Safety, Volume 213, 2021.

[8] "MPU-6000/MPU-6050 Product Specification" InvenSense Ltd. 8, 2013.

[9] G. Welch and G. Bishop, "An Introduction to the Kalman Filter," UNCChapel Hill, TR 95-041, July 24, 2006.

[10] Pengfei Gui, Liqiong Tang, Subhas Mukhopadhyay "MEMS Based IMU for Tilting Measurement: Comparison of Complementary an Kalman Filter Based Data Fusion" Auckland, New Zealand, 15-17 June 2015.

[11] "ESP32 Series Datasheet" Version 3.7 Espressif Systems https://www.espressif.com/

[12] G. Tevatia and S. Schaal, "Inverse kinematics for humanoid robots," Proceedings of IEEE International Conference on Robotics and Automation. Symposia Proceedings, Vol. 1, pp. 294-299, 2000.

[13] O. Carbajal-Espinosa, F. Izar-Bonilla, M. D´ıaz-Rodr´ıguez and E. Bayro-Corrochano "Inverse Kinematics of a 3 DOF Parallel Manipulator: A Conformal Geometric Algebra Approach" Cancun, Mexico, Nov 15-17, 2016

[14] DongGyu Kim, Sung-Ho Hwang*, "Kinematic Implementation of 3-DOF 2-Link Type Vehicle Simulator" Korea, December 19,2020

[15] Hang Wang[1,2], Hanghang Qi[3], Minghui Xu[3], Yanhua Tang[4], Jiantao Yao[5,6], Xuedong Yan[7,8], Ming Li[2] "Research on the Relationship between Classic Denavit-Hartenberg and Modified Denavit-Hartenberg" 2014

[16] Kiam Heong Ang, Gregory Chong, "PID Control System Analysis, Design, and Technology" 4, July 2005