Center of Pressure Control for Balancing Humanoid Dance Robot Using Load Cell Sensor, Kalman Filter and PID Controller

Cisi Fitri Wulandari [1], Abdul Fadlil [2,*]
[1,2] Department of Electrical Engineering, Universitas Ahmad Dahlan, Yogyakarta, Indonesia
Email: [1] wulandarifitricisi@gmail.com, [2] fadlil@mti.uad.ac.id
*Corresponding Author

***Abstract*— Balance control on the Lanage Jagad humanoid dance robot is one of the means to create flexible dance movements in the robot movement system to make it more stable and can reduce the frequency of the robot falling or being unable to maintain balance when performing the dance. For the position of the robot, it can use a weight sensor or load cell sensor, the sensor measures the resistance value that can control the weight of 4 weight points on each robot leg which will later be converted into a pressure value at each point, in the study. This test was carried out with the same control behavior using an inertial sensor MPU6050. The balance on the robot uses a balance based on a load cell, which is a situation where the position of the robot in coordinates approaches the center of balance or CoP (Center of Pressure) at coordinates (0,0) or if using MPU6050 it is in a far error value condition so that it can balance the conditions so as not to falls close to the value of the robot state based on ZMP (Zero Moment Point) and CoG (Central of Gravity) as the MPU6050 sensor placement. In this study, for the balance control system using the Arduino MEGA 2560 PRO Board as a complement to the OpenCM 9.04 microcontroller, using 8 load cell sensors to determine the balance point which has been made predictions of pressure from the load cell using a kalman filter also PID control to handle the servo motor. The results from the center point of the robot's pressure have succeeded in determining the center of balance or CoP based on the X coordinates of 0 and the Y coordinates of 0 and the quadrant direction based on the center of gravity, so that the results of the balance system in standing and dancing conditions are based on the center of balance using a load cell, the success rate when standing by 87.5% and balance when dancing by 89%.**

Keywords*— *Humanoid Robot, Kalman Filter, Center of Pressure, PID Control, Load Cell

I. INTRODUCTION

In the joining national robot dance competition in Indonesia, KRSTI (Kontes Robot Seni Tari Indonesia), the robot must be able to dance by following the music according to the dance theme, therefore dance movements with music must be in harmony. The robot will start dancing when the music is turned on and will stop dancing when the music is turned off. The KRSTI competition arena consists of 3 zones, namely zones A, B, and C. Zone A is the first zone with obstacles, namely the robot doing the opening movement. Zone B is a zone that must be taken after passing zone A with the provision that the robot must be able to perform the core dance movements according to a predetermined theme every year until the zone ends. Zone C is the last zone for the robot to complete dance moves, in zone C the robot must be able to perform closing movements from the specified dance theme. High flexibility and balance are required by dance robots in performing dance movements. The balance of the robot will determine the value obtained during the game. If the robot is unbalanced and falls, it will get a retry and a reduction in value. The condition of the field of the match track is not always flat and some dance movements require the balance of the dance robot to keep the robot from falling. Humanoid Dance Robot often when performing dance movements often has trouble, namely losing balance when doing dance movements. This is because the humanoid dance robot is still having problems in maintaining balance in performing the dance.

In the previous Lanange Jagad humanoid dance robot, research on balance control has been carried out by [1] about use of Gyroscope and Accelerometer Sensors in the Balance System of the Lanange Jagad Humanoid Dance Robot." The research was conducted to test the balance system of the Lanange Jagad humanoid dance robot on a track plane slope with an inclination of 0° to 10° so that it does not fall forward and backward. This is a challenge for the Lanange Jagad team to be able to maintain balance so as not to fall to the right or left sides. Therefore, we propose project entitled Center of Pressure for Balancing Lanange Jagad Humanoid Dance Robot Using Load Cell Sensor. We hope that this research can improve stability in maintaining the balance of the right and left sides.

In this study, for the balance control system using the Arduino MEGA 2560 PRO Board as a complement to the OpenCM9.04 microcontrollers and the MPU6050 sensor which has acceleration values in 3 axes or accelerometer which has units (m/s²) and angular velocity or gyroscope which has units (rad). /s) to signal that the robot is in a stable state or in Zero Moment Point (ZMP) [2]. On each left foot and right foot, 4 load cell sensors are installed on the 4 outermost points of the right foot and 4 outermost points of the left foot which are converted via 8 HX711 modules using ADC based on the same time on Arduino Mega 2560 PRO as control system. Each sensor MPU6050 and load cell for balance control reference uses a sensor reading estimation system in the form of a Kalman Filter [3]. After the balance determination process is carried out using PID control [4], the servo actuators are ID 19, 20, 21, and 22 as front and rear balance feedback, and ID 15,16, 23, and 24 as right and left balance feedback for the robot. , so that it will carry out the

reciprocal accumulation of the predetermined balance center point. So that the robot can maintain balance from the right side and the left side so as not to fall. The balance on the robot is a situation where the position of the robot in the coordinates is close to (0,0) or is in a condition where the error value is far so that it can balance the conditions so as not to fall. At coordinates that are close to (0,0) the robot is in the condition of the center of mass and gravity of the robot. The coordinates of the robot's center of pressure can be used to determine the direction of feedback from the error value generated by changes in load when the robot does standing and dance movements on an uneven track.

II. METHOD

A. Balancing System

The *Lanange Jagad* humanoid dance robot system uses the arduino MEGA 2560 PRO as the main controller board. The dynamixel servo can be accessed using the OpenCM9.04 board communicated from the Arduino MEGA 2560 PRO. The MPU6050 gyroscope and accelerometer sensors and load cell sensors will be connected to the Arduino MEGA 2560 PRO. The MPU6050 sensor to compare the balance uses a load cell sensor to obtain a balance to the maximum of the robot's ability. The hardware block diagram of the *Lanange Jagad* humanoid dance robot is shown in Fig. 1.

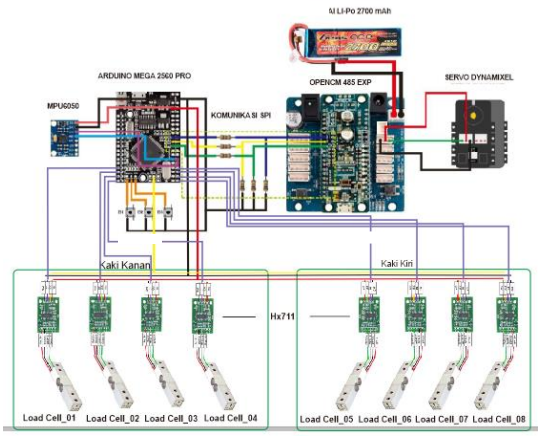

Fig. 1. Balancing Hardware System Block Diagram

The flowchart of the balance system program can be seen in the flowchart Fig. 2, there are five modes, among others, the first modes one to three explain the balance that is controlled based on the MPU6050 sensor and modes four to five are explained about the balance that is controlled based on the load cell sensor.

The mode change instructions are carried out using the mode button one for stability when the robot is stationary, the mode button two for instructing the robot to stabilize when standing based on the MPU6050 sensor, the mode button three for instructing the robot to stabilize when the robot dances based on the MPU6050 sensor, the mode button four for instructing the robot to perform stability when standing based on the load cell sensor, and five mode buttons to instruct the robot to stabilize when dancing based on the load cell sensor. The program for each button will perform a looping process before the reset button is pressed, which indicates the program has been run.

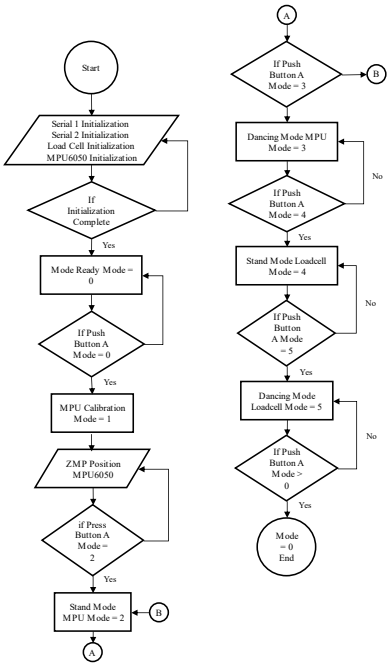

Fig. 2. Flowchart Balancing System

B. Load Cell Measurement to Newton

Determination of Center of Gravity (CoG) [5] [6] has been carried out by determining the coordinate value (x, y), then in the application of the center point of the robot pressure using a load cell which is a sensor for detecting the weight of objects, then to convert from weight to pressure in units (N) newtons, it can be seen in equation (1).

$$F = ma \tag{1}$$

C. Center of Pressure (CoP)

Center of Pressure (CoP) requires construction and parameters according to the method used where, in the construction of the robot leg there are 4 load cell sensors on each robot leg located at each corner of the robot foot, can be seen in Fig. 3.

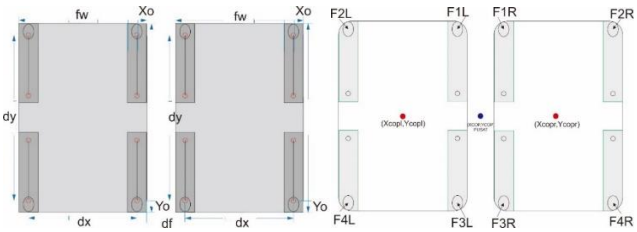

Fig. 3. Variable Foot Measurement of CoP

The center of the robot foot pressure is obtained by calculating the pressure value for each foot, then the pressure value is calculated to obtain the center coordinates of each left foot and right foot using equations (2) to (3). The equation variable can be seen in Fig. 3 with a value that corresponds to the size of the Lanange Jagad humanoid dance robot.

$$Xcop = Xo + \frac{(F_{2l} + F_{4l} + F_{2r} + F_{4r}) * dx}{F_{total}} + \frac{(F_{1r} + F_{2r} + F_{3r} + F_{4r}) * (fw + df)}{F_{total}} \tag{2}$$

$$Ycop = Yo \frac{(F_{1l} + F_{2l} + F_{1r} + F_{2r}) * dy}{F_{total}} \quad (3)$$

Where $Xcop$ is the X-coordinate at the center of pressure of the robot's footwear, $Ycop$ is the Y coordinate at the center of pressure of the robot's footwear, fw is foot width, df is the distance between the right and left legs, Xo is the distance of the $F2$ load cell pressure point to the nearest long side of the footwear, Yo is the distance of the $F4$ load cell pressure point to the nearest wide side of the shoe, dx is the distance from the $F1$ load cell pressure point to the $F2$ load cell pressure point (horizontally), dy is the distance of the $F1$ load cell pressure point to the $F3$ load cell pressure point (Vertical), F_{1r} is the Pressure Force on the load cell 1 foot right, F_{1l} is the Pressure Force on the load cell 1 foot left, F_{2r} is the Pressure Force on the load cell 2 feet right, F_{2l} is the Pressure Force on the load cell 2 feet left, F_{3r} is the Pressure Force on the load cell 3 feet right, F_{3l} is the Pressure Force on the load cell 3 feet left, F_{4r} is the Pressure Force on the load cell 4 feet right, F_{4l} is the Pressure Force on the load cell 4 feet left, Ftotal is the total Pressure Force.

The addition of Xstate and Ystate as a modification of the center of balance becomes consistent when getting the same weight on each leg. The equation is contained in equations (4) to (5).

$$Xstate = (fw + \frac{df}{2}) \quad (4)$$

$$Ystate = (\frac{dy + Yo * 2}{2}) \quad (5)$$

Where $Xstate$ is the control of the position of the X coordinate to the result of the original center of pressure and $Ystate$ is the control of the position of the Y coordinate to the result of the original center of pressure

The results of the modification of the addition of Xstate and Ystate as a modification of the center of balance become consistent when getting the same weight on each leg. The equation is contained in equations (6) to (7).

$$Xcop = Xo + \frac{(F_{2l} + F_{4l} + F_{2r} + F_{4r}) * dx}{F_{total}} + \frac{(F_{1r} + F_{2r} + F_{3r} + F_{4r}) * (fw + df)}{F_{total}} - (fw + \frac{df}{2}) \quad (6)$$

$$Ycop = Yo \frac{(F_{1l} + F_{2l} + F_{1r} + F_{2r}) * dy}{F_{total}} - (\frac{dy + Yo * 2}{2}) \quad (7)$$

D. Kalman Filter

Kalman filter is an algorithm that functions as a counterweight to statistical noise and uncertainty over time, resulting in a more reliable estimate of object status than a single measurement. Kalman filter as a balance filter algorithm has two steps, namely predicting to produce an estimate of the current state variable along with its uncertainty, and namely updating taking a set of measurements to compare with the predicted data and updating the estimate with an average coefficient called kalman gain [7]. The kalman filter equation can be written as.

$$\hat{x}_{t|t-1} = F_t \hat{x}_{t-1|t-1} + B_t u_t \quad (8)$$

$$P_{t|t-1} = F_t P_{t-1|t-1} F_t^T + Q_t \quad (9)$$

$$\hat{x}_{t|t} = \hat{x}_{t|t-1} + K_t(y_t - H_t \hat{x}_{t|t-1}) \quad (10)$$

$$K_t = P_{t|t-1} H_t^T (H_t P_{t|t-1} H_t^T + R_t)^{-1} \quad (11)$$

$$P_{t|t} = (1 - K_t H_t) P_{t|t-1} \quad (12)$$

Where $\hat{x}$ is an estimate on prediction, F is the transition matrix on prediction, u is the control variable, B is the control matrix, P is the estimate of the prediction error, Q is an error that occurs in the process, y is the measurement variable, H is a matrix measurement, K is Kalman gain, R is the prediction error of the measurement, t|t-1 is the measurement period [8-10].

In this study using 2 different kalman filter methods based on the use of each sensor. The kalman filter method used to estimate the MPU6050 sensor [11-13] is found in equation (8)-(12). The kalman filter method on the MPU6050 is used to compare the balance system using a load cell [14-18]. The kalman filter method used to estimate the load cell sensor is in equation (13)-(17).

$$x_{t|t-1} = x_{t-1|t-1} \quad (13)$$

$$P_{t-1} = P_{t-1|t-1} + Q_t \quad (14)$$

$$x_{t|t} = x_{t|t-1} + K_t(y_t - x_{t|t-1}) \quad (15)$$

$$K_t = P_{t|t-1} (P_{t|t-1} + R)^{-1} \quad (16)$$

$$P_{t|t} = (1 - K_t) P_{t|t-1} \quad (17)$$

E. PID Servo Controller

The balance control system for the Lanange Jagad humanoid dance robot goes through several stages, including through the process of collecting data from the load cell sensor or MPU6050 according to the button conditions described in the flowchart diagram Fig. 2, then the estimation process is carried out using a kalman filter, then the control process PID to give the effect of sensitivity of each joint of the robot. Basically, the simple of equation control PID using equations [19]. From basically equation not compatible with 3 axis of foot servo, and we were modified the equation to value of *P* is proportional control, *I* is integral control, and *D* is Derivative control get from equations (18) – (20) [4]. Balancing controller MPU6050 block diagram in Fig. 4.

$$P = error * KP \quad (18)$$

$$I = (last\_integral * KI)/TI \quad (19)$$

$$D = (Error - Last\_Error) * KD \quad (20)$$

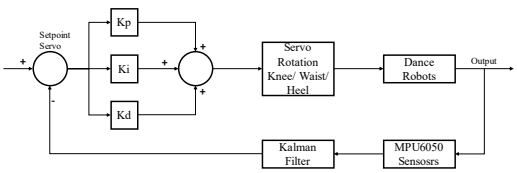

Fig. 4. Balancing controller MPU6050 block diagram

Balance control using a MPU6050 sensor like Fig. 4 begins including through the process of collecting data from the MPU6050 according to the button conditions described in the flowchart diagram Fig. 2, then the estimation process is carried out using a kalman filter, then the control process PID to give effect to the sensitivity of each joint of the robot. Balancing controller load cell block diagram shown in Fig. 5.

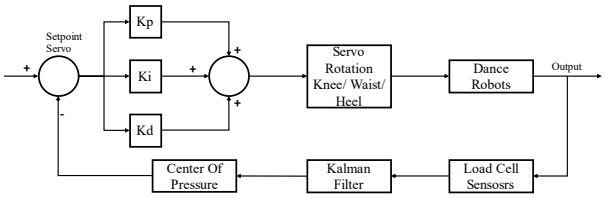

Fig. 5. Balancing controller load cell block diagram

Balance control using a load cell sensor like Fig. 5 begins with a set point which has an initial value of 2048 servo rotation, the initial value is the initial value of the servo rotation before being influenced by the condition of the robot in maintaining balance. The rotary value that has been influenced by PID control produces a new rotary value, then the rotary servo robot changes according to the condition of the robot based on the load cell sensor which has been estimated by the kalman filter and has been measured based on CoP.

III. Result and Discussion

A. Sensor Testing

The weight test data that has been tested on each load cell on the right leg can be seen in Table 1. The error obtained using equation (2.8) at the sensor point of the load cell F2r is 0.453, F4r is 0.766, F3r is 0.328, and F1r is 0.492. The results of the average error generated indicate that the load cell sensor has high accuracy on the right leg.

Table 1. Right Foot Load Cell Sensor Test

Digital Scales Masa	Measured Time (gram)				Error (gram)			
	F2r	F4r	F3r	F1r	F2r	F4r	F3r	F1r
53	53	53	53	53	0.05	0.01	0.04	0.06
105	106	106	106	105	1.25	0.8	0.58	0.46
146	147	147	147	147	0.7	0.67	0.8	0.5
224	224	225	224	224	0.47	1.35	0.3	0.48
258	259	260	258	258	0.82	2.3	0.46	0.06
252	252	252	253	252	0.32	0.01	0.55	0.02
322	322	323	322	323	0.08	0.58	0.04	0.5
429	429	429	429	430	0.04	0.05	0.07	0.8
464	464	465	464	465	0.3	0.86	0.4	0.8
542	543	543	542	543	0.5	0.85	0.04	1.24
Average error H					0.453	0.766	0.328	0.492

The weight test data that has been tested on each load cell on the left leg can be seen in Table 2. The error obtained using equation (2.7) at the sensor point of the load cell F2r is 0.384, F4r is 0.609, F3r is 0.674, and F1r is 0.408. The results of the average error generated indicate that the load cell sensor has high accuracy on the left foot.

B. Kalman Filter Estimation Testing

In Fig. 6 diagram, there are 2 blue lines which are the original sensor data and the red line is the estimated data. Where the blue line uses a variable R of 3 and Q of 0.1. This shows that the kalman filter is suitable for use for load cells in estimating sensor values that have a lot of potential for interference, so the results obtained are that the balance will be softer because the sensor value of the load cell is more stable [20].

Table 2. Left Foot Load Cell Sensor Test

Digital Scales Mass	Measured Time (gram)				Error (gram)			
	F2l	F4l	F3l	F1l	F2l	F4l	F3l	F1l
53	54	53	54	53	0.63	0.35	0.86	0.06
105	105	106	105	105	0.25	0.5	0.07	0.46
146	146	147	147	146	0.26	0.84	0.66	0.06
224	224	225	225	224	0.26	0.98	1.46	0.48
258	259	259	259	258	0.58	0.68	0.98	0.06
252	252	253	253	253	0.05	0.86	0.95	1.2
322	322	322	323	322	0.25	0.02	0.5	0.3
429	429	429	429	429	0.02	0.3	0.46	0.4
464	465	465	464	464	0.78	0.58	0.46	0.08
542	543	543	542	543	0.76	0.98	0.34	0.98
Average error H					0.384	0.609	0.674	0.408

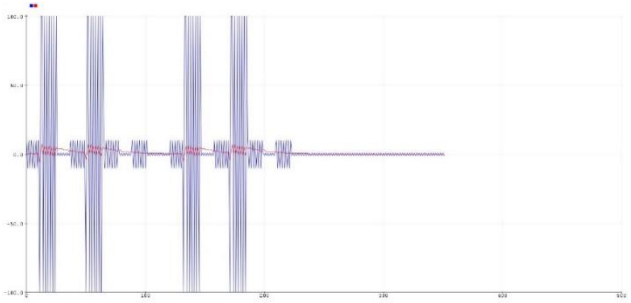

Fig. 6. Kalman Filter Estimation Results on Load cell

C. PID Controller Testing

Fig. 7, 8, 9 show 3 times the PID oscillation test which can be seen in Table 2 with variations in color and values of KP, KD, KI, TI, and (Imax) is control of integral maximum value. Variable of PID Controller can be seen in Table 3. The first test has a very high oscillation value compared to the blue and green waves. The second test has an oscillation value that has a value that is not too low and not too high or is in the middle value of the red and green waves. The third test has a very low oscillation value compared to the blue and blue waves.

Table 3. Variable of PID Controller

Color	KP	KD	KI	TI	Imax	Servo
Red	5.0	2.6	20.0	25	200	Heel
Blue	1.0	1.8	10.0	25	200	Heel
Green	0.5	0.6	10.0	25	200	Heel
Red	5.0	2.6	20.0	20	200	knee
Blue	1.0	1.8	15.0	20	200	knee
Green	0.5	0.6	10.0	20	200	knee
Red	0.8	21.6	52.0	20	150	waist
Blue	4.0	10.8	26.0	20	150	waist
Green	4.0	10.8	26.0	20	150	waist

D. Center of Pressure Testing

Testing the center point of the robot's pressure in Fig. 10 placing the load cell sensor on the foot of the foot frame so that when the robot is standing the direction of the pressure received by the load cell is towards the top, the experiment is carried out using a simulated variable value in order to ensure that the difference in modifications from the formula that has been previously applied by adding Xstate and Ystate to get a value close to the coordinates (0,0).

In testing the center point of the robot's quadrant pressure Fig. 11 which shows the direction of the pressure center coordinates based on the robot's center of pressure quadrant, the points and directions of the red line show data from the robot's left slope table, the green line is the front slope, the yellow color is the back slope and the color blue is the right slope, and the pink list shows the data does not match the quadrant division rule or the robot's pressure direction based on the center of the pressure point quadrant.

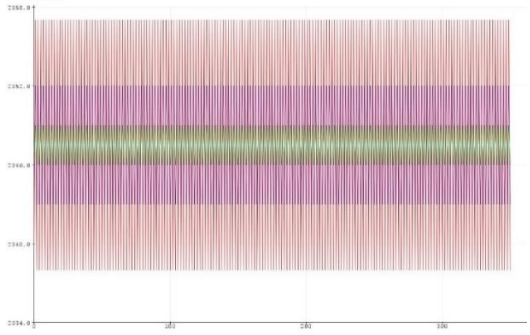

Fig. 7. PID Heel Servo

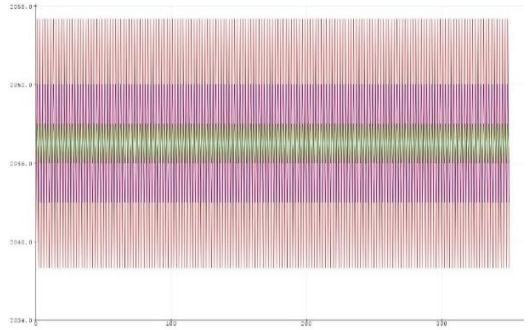

Fig. 8. PID Knee Servo

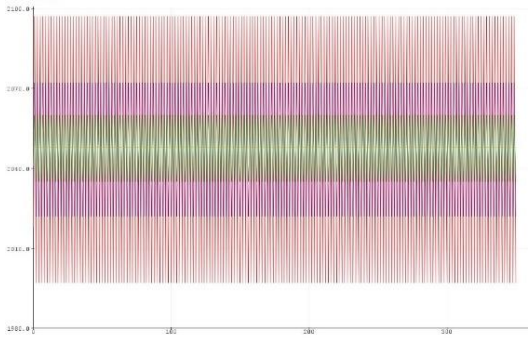

Fig. 9. PID Waist Servo

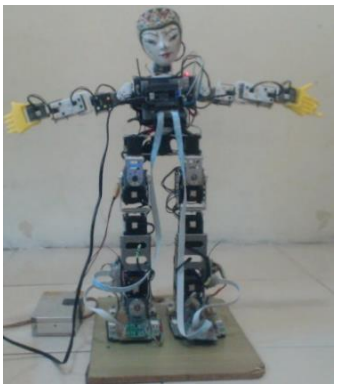

Fig. 10. Load Cell Position on Robot Legs

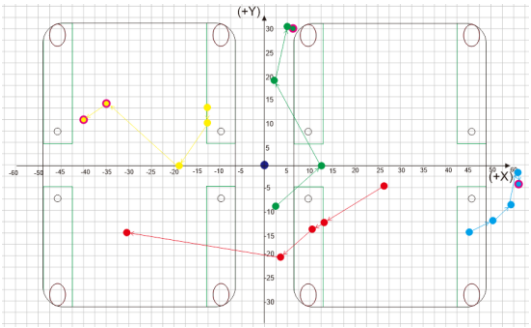

Fig. 11. Robotic Quadrant Center Pressure Point Test

E. Balancing Robot Testing

In Table 4 are the results of testing the balance of a standing robot with a track slope to the right at an angle of 5°, 8°,11° 15°¸ 20°, 25° When using the load cell balance system the robot does not fall while with the MPU6050 balance system at an angle of 8°. Does not fall and without balance the robot falls more than 11°.

Table 5 shows the results of testing the balance of a standing robot with a track tilt to the left at an angle of 5°, 8°,11° 15°¸ 20°,25°. When using the load cell balance system, the robot does not fall while with the MPU6050 balance system at an angle of 8°, it does not fall and without balance the robot falls more than 8°.

In Table 6 are the results of testing the balance of a standing robot with a track slope to the left at an angle of 5°, 8°,11° 15°¸ 20°,25°. When using the load cell balance system, the robot does not fall while with the MPU6050 balance system at an angle of 5°, 8°,11° it does not fall and without balance the robot falls.

In Table 7 are the results of testing the balance of a standing robot with a track slope to the left at an angle of 5°, 8°,11° 15°¸ 20°,25°. When using the load cell balance system, the robot does not fall while with the MPU6050 balance system at an angle of 5°, 8°,11° it does not fall and without balance the robot falls more than 11°.

IV. CONCLUSION

The research that has been realized can be concluded that the robot has successfully designed a balance system for the KRSTI Lanange Jagad humanoid dance robot using a load cell sensor to maintain the balance of the robot. Second, the robot has succeeded in determining the center of balance based on the X coordinates of 0 and the Y coordinates of 0 and the direction of the quadrant based on the center of gravity. Third, the robot has succeeded in making a balance system in conditions based on the center of balance using a load cell, the success rate when standing is 87.5%, and balance when dancing is 89%. In the research that has been done, it is found that there are deficiencies that can be developed in the future, as for suggestions for development, namely the use of clocks used for eight load cells in order to maximize reading of load cell sensors, as well as designing hardware that can minimize load cell sensors so that the robot can move freely. Actually, load cell using kalman filter because just using one clock pin of 8 input data, it is so high frequency of the jumping data or high oscillation. In the future, we want to recommendation using 8 pin clock for ones.

Table 4. Right Inclination Standing Balance Test

angle (°) / to	Not Use Balancing							Balance MPU6050							Balance Load Cell						
	1	2	3	4	5	6	7	1	2	3	4	5	6	7	1	2	3	4	5	6	7
5	X	X	√	X	√	X	X	√	√	√	√	√	X	√	√	√	√	√	√	√	√
8	X	X	X	√	√	X	X	X	X	√	√	√	√	√	√	√	√	√	√	√	√
11	X	X	X	X	X	X	X	X	X	X	X	X	X	X	√	√	√	X	√	√	√
15	X	X	X	X	X	X	X	X	X	X	X	X	X	X	√	√	X	√	√	X	√
20	X	X	X	X	X	X	X	X	X	X	X	X	X	X	√	√	√	X	√	√	√
25	X	X	X	X	X	X	X	X	X	X	X	X	X	X	√	√	X	√	√	√	X

Table 5. Left Slope Standing Balance Test

angle (°) / to	Not Use Balancing							Balance MPU6050							Balance Load Cell						
	1	2	3	4	5	6	7	1	2	3	4	5	6	7	1	2	3	4	5	6	7
5	X	X	√	X	X	√	X	√	√	X	√	√	X	√	√	√	√	√	√	√	√
8	X	X	X	√	X	√	X	X	√	√	√	X	√	√	√	√	√	√	√	√	√
11	X	X	X	X	X	X	X	X	X	X	X	X	X	X	√	√	X	√	X	√	√
15	X	X	X	X	X	X	X	X	X	X	X	X	X	X	√	√	√	√	X	√	√
20	X	X	X	X	X	X	X	X	X	X	X	X	X	X	√	√	X	√	√	√	X
25	X	X	X	X	X	X	X	X	X	X	X	X	X	X	√	√	√	√	√	X	√

Table 6. Front Slope Standing Balance Test

angle (°) / to	Not Use Balancing							Balance MPU6050							Balance Load Cell						
	1	2	3	4	5	6	7	1	2	3	4	5	6	7	1	2	3	4	5	6	7
5	X	X	√	X	X	√	X	√	√	X	√	√	X	√	√	√	√	√	√	√	√
8	X	X	X	√	X	X	X	X	√	√	√	X	√	√	√	√	√	√	√	√	√
11	X	X	X	X	X	X	X	√	√	√	√	X	X	X	√	√	√	√	√	√	√
15	X	X	X	X	X	X	X	X	X	X	X	X	X	X	√	√	√	√	√	√	√
20	X	X	X	X	X	X	X	X	X	X	X	X	X	X	√	X	√	√	√	√	X
25	X	X	X	X	X	X	X	X	X	X	X	X	X	X	√	√	X	√	√	√	√

Table 7. Rear Tilt Standing Balance Test

angle (°) / to	Not Use Balancing							Balance MPU6050							Balance Load Cell						
	1	2	3	4	5	6	7	1	2	3	4	5	6	7	1	2	3	4	5	6	7
5	X	X	√	X	X	√	X	√	√	X	√	√	X	√	√	√	√	√	√	√	√
8	X	X	X	√	X	√	X	X	√	√	√	X	√	√	√	√	√	√	√	√	√
11	X	X	X	X	X	X	X	√	√	√	√	X	X	X	√	√	X	√	√	√	√
15	X	X	X	X	X	X	X	X	X	X	X	X	X	X	√	X	√	√	√	√	√
20	X	X	X	X	X	X	X	X	X	X	X	X	X	X	√	√	X	√	√	√	X
25	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	√	√	√	√	√	X

REFERENCES

[1] A. S. Samosir and N. S. Widodo, "Gyroscope and Accelerometer Sensor on the Lanange Jagad Dance Robot Balance System," *Buletin Ilmiah Sarjana Teknik Elektro*, vol. 2, no. 2, pp. 51–58, 2020, https://doi.org/10.12928/biste.v2i2.922.

[2] I. Rifajar and A. Fadlil, "The Path Direction Control System for Lanange Jagad Dance Robot Using the MPU6050 Gyroscope Sensor," *International Journal of Robotics and Control Systems*, vol. 1, no. 1, pp. 27–40, 2021, https://doi.org/10.31763/ijrcs.v1i1.225.

[3] A. Nawrocka, M. Nawrocki, and A. Kot, "The use of Kalman Filter in Control the Balancing Robot," *Proceedings of the 2020 21st International Carpathian Control Conference*, 2020, https://doi.org/10.1109/ICCC49264.2020.9257212.

[4] K. Joni, M. Ulum, T. Prasetyo, and A. Y. Maulana, "Dynamic balancing humanoid robot using completmentary filter to optimized pid controller," *Journal of Physics: Conference Series*, vol. 1211, no. 1, p. 012046, 2019, https://doi.org/10.1088/1742-6596/1211/1/012046.

[5] A. H. Alasiry, N. F. Satria and A. Sugiarto, "Balance Control of Humanoid Dancing Robot ERISA while Walking on Sloped Surface using PID," *2018 International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)*, pp. 577-581, 2018, https://doi.org/10.1109/ISRITI.2018.8864447.

[6] S. Kim, K. Hirota, T. Nozaki and T. Murakami, "Human Motion Analysis and Its Application to Walking Stabilization with COG and ZMP," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 11, pp. 5178-5186, 2018, https://doi.org/10.1109/TII.2018.2830341.

[7] R. Subburaman, D. Kanoulas, L. Muratore, N. G. Tsagarakis, and J. Lee, "Human inspired fall prediction method for humanoid robots," Robotics and Autonomous Systems, vol. 121, p. 103257, 2019, https://doi.org/10.1016/j.robot.2019.103257.

[8] M. S. Grewal, A. P. Andrews, C. G. Bartone, "Kalman Filtering," *Global Navigation Satellite Systems, Inertial Navigation, and Integration*, pp.355-417, 2020, https://doi.org/10.1002/9781119547860.ch10.

[9] Y. Xu, K. Xu, J. Wan, Z. Xiong and Y. Li, "Research on Particle Filter Tracking Method Based on Kalman Filter," *2018 2nd IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC)*, pp. 1564-1568, 2018, https://doi.org/10.1109/IMCEC.2018.8469578.

[10] G. Revach, N. Shlezinger, X. Ni, A. L. Escoriza, R. J. G. van Sloun and Y. C. Eldar, "KalmanNet: Neural Network Aided Kalman Filtering for Partially Known Dynamics," *IEEE Transactions on Signal Processing*, vol. 70, pp. 1532-1547, 2022, https://doi.org/10.1109/TSP.2022.3158588.

[11] A. A. Rafiq, W. N. Rohman, S. D. Riyanto, "Development of a simple and low-cost smartphone gimbal with MPU-6050 sensor," *Journal of Robotics and Control (JRC)*, vol. 1, no. 4, pp. 136-140, 2020, https://doi.org/10.18196/jrc.1428.

[12] I. Rifajar, A. Fadlil, "The Path Direction Control System for Lanange Jagad Dance Robot Using the MPU6050 Gyroscope Sensor," *International Journal of Robotics and Control Systems*, vol. 1, no. 1, pp. 27-40, 2021, https://doi.org/10.31763/ijrcs.v1i1.225.

[13] H. Heriyadi, H. Fajrin, and W. Kartika, "Prayer Guide Gyroscope Bracelet for The Deaf Using MPU6050 Sensor", *Indonesian Journal of Electronics, Electromedical Engineering, and Medical Informatics*, vol. 4, no. 1, pp. 36-40, 2022, https://doi.org/10.35882/ijeeemi.v4i1.6.

[14] L. Yang, H. Yao, J. Wang, C. Jiang, A. Benslimane and Y. Liu, "Multi-UAV-Enabled Load-Balance Mobile-Edge Computing for IoT

Networks," *IEEE Internet of Things* Journal, vol. 7, no. 8, pp. 6898-6908, 2020, https://doi.org/10.1109/JIOT.2020.2971645.

[15] C. U. Ebuzeme, Z. A. Quadri, O. Noah, E. O. Ogedengbe, C. Eguma, "Performance Evaluation of an Aerodynamic Blade Model Using a Bottom-mounted Force Balance System," *AIAA Propulsion and Energy 2019 Forum*, p. 4160, 2019, https://doi.org/10.2514/6.2019-4160.

[16] B. A. Alqahtani, *et al.*, "Effect of community-based group exercise interventions on Standing balance and strength in independent living older adults," Journal of geriatric physical therapy, vol. 42, no. 4, pp. E7-E15, 2019, https://doi.org/10.1519/JPT.0000000000000221.

[17] A. A. Neghabi, N. Jafari Navimipour, M. Hosseinzadeh and A. Rezaee, "Load Balancing Mechanisms in the Software Defined Networks: A Systematic and Comprehensive Review of the Literature," *IEEE Access*, vol. 6, pp. 14159-14178, 2018, https://doi.org/10.1109/ACCESS.2018.2805842.

[18] T. V. Chien, E. Björnson and E. G. Larsson, "Optimal Design of Energy-Efficient Cell-Free Massive Mimo: Joint Power Allocation and Load Balancing," *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5145-5149, 2020, https://doi.org/10.1109/ICASSP40776.2020.9054083.

[19] D. Saputra, A. Ma'arif, H. Maghfiroh, P. Chotikunnan, S. N. Rahmadhia, "Design and Application of PLC-based Speed Control for DC Motor Using PID with Identification System and MATLAB Tuner," *International Journal of Robotics and Control Systems*, vol. 3, no. 2, pp. 233-244, 2023, https://doi.org/10.31763/ijrcs.v3i2.775.

[20] H. Fang, N. Tian, Y. Wang, M. Zhou and M. A. Haile, "Nonlinear Bayesian estimation: from Kalman filtering to a broader horizon," *IEEE/CAA Journal of Automatica Sinica*, vol. 5, no. 2, pp. 401-417, 2018, https://doi.org/10.1109/JAS.2017.7510808.