XLVI CILAMCE
VITÓRIA - ES - BRAZIL - 2025
XLVI IBERO-LATIN AMERICAN CONGRESS ON COMPUTATIONAL METHODS IN ENGENEERING

Evaluation of the Analytical Modeling of a Stewart-Gough Actuated Ball-and-Plate System using Experimental Data

Flavio C. A. Pereira[1], William R. Lüdtke[1], João V. C. Fontes[1], Giuliana S. Venter[1]

[1]*Dept. of Mechanical Engineering, Federal University of Paraná (UFPR)*
Av. Cel. Francisco H. dos Santos, 230, Block IV, 81530-001, Curitiba, Paraná, Brazil
flaviocanfilde@ufpr.br, william.ludtke@ufpr.br, giuliana.venter@ufpr.br, fontes.joao@ufpr.br

Abstract. Stewart-Gough Platforms (SGPs) provide precise 6-degree-of-freedom control, useful for applications like offshore wave stabilization. Coupling an SGP with a ball-and-plate system (BPS) creates a testbed for nonlinear trajectory control. In this work, an analytical model was developed to simulate the behavior a physical SGP-BPS system. The finished model is based on principles of analytical geometry, kinematics and dynamics. The physical platform used six servomotors to orientate a plate with a touchscreen sensor. The position of a free-rolling steel sphere on the plate was tracked by sensors, while a microcontroller calculated and sent servo angles to adjust the trajectory of the sphere. Validation used data from 90 experiments recording sphere position, timestamps, and actuator angles across various scenarios. The model, given initial sphere position and actuator angles over time, predicted plate orientation and sphere position/velocity. Its accuracy was assessed by comparing simulated versus experimental sphere positions, primarily using the Euclidean norm error over time. Results showed preliminary correlation with notable residual errors. The validated model can be further used for creating a digital twin for real-time control and developing data-driven controllers for offshore stabilization systems.

Keywords: Analytical modeling, Experimental validation, Stewart platform, Kinematics, Parallel manipulators.

1 Introduction

A generalized parallel manipulator is defined by Merlet [1] as a closed-loop kinematic chain mechanism that has the end-effector linked to the base by independent kinematic chains, and a fully parallel manipulator as consisting of a moving platform with n degrees of freedom connected to the base by n independent kinematic chains. According to Silva et al. [2], this concept is advantageous when compared to serial robots, due to it's higher acceleration, motion accuracy and load capacity (as several parallel chains are supporting the object's weight).

The most widely known parallel robot is the Stewart-Gough Platform (SGP), introduced by Gough [3] for tire testing applications and independently by Stewart [4] as a mechanism for flight simulators. The proposed model consisted of a 6-degree-of-freedom (6-DOF) system actuated by 6 closed-loop kinematic chains. Nowadays, the SGP has a wide application across several industries, such as flight simulators, spacecraft docking, astronomical telescopes, positioning tools and milling machines (Xu et al. [5] and Borras et al. [6]). A notable application presented in Li et al. [7] is the wave motion compensation for offshore operations that are critically dependent on sea conditions, including inter-vessel cargo transfer and oil and gas exploration activities. A ball-and-plate system (BPS) can be associated with the SGP, providing an interesting problem for nonlinear control. The BPS, as described in Kassem et al. [8], is an extension of the 2-DOF ball and beam system. In the BPS, a ball can freely roll over a plate, which increases the complexity, as the system now has 4-DOF. The system becomes even more complex when the plate of the BPS is attached to the end effector of an SGP, creating a nonlinear SGP-BPS system.

Given the wide applicability of the SGP and the potential of the SGP-BPS as a testbed for nonlinear control strategies, it is desirable to develop both numerical and physical representations of these systems for controller development and validation. The Federal University of Paraná (UFPR) currently holds a physical SGP-BPS at the Laboratory of Monitoring and Control (LaMoC) from the Mechanical Engineering Department. However, as pointed out by Kang et al. [9], deploying controllers directly to physical systems without assessing their feasibility can cause unexpected behaviors and safety risks. It is thus preferable to use a numerical model initially, to detect possible anomalies, and transfer the controller to the physical system afterwards. Numerical models can also be useful in the development of digital twins, as in Kang et al. [9] and Zhang et al. [10]. Due to these reasons, this work developed an analytical model of the SGP-BPS system, using the physical system as a means of validation

of this model. This involved the acquisition of data from the SGP-BPS, the formulation of the SGP-BPS analytical model, and the evaluation of the model using the acquired data. The developed model aims to determine the position of the ball on the plate over time as a function of the angles of the SGP servomotors.

2 Methodology

The physical SGP-BPS system used in this work was developed and manufactured by Vieira [11], and is presented on Fig. 1. The main electronic components of the system are six MG90D servo motors, one PCA9085 servo driver board, one 4-wire resistive touchscreen sensor, one Arduino UNO R3 microcontroller board, and a metallic ball. The data used for this work was acquired by Lüdtke [12] according to the methodology presented in Lüdtke and Venter [13]. Inputs for the microcontroller are the desired position of the ball (target) and the actual position of the ball (acquired from the touchscreen sensor), whilst the outputs are the desired positions of each servo actuator.

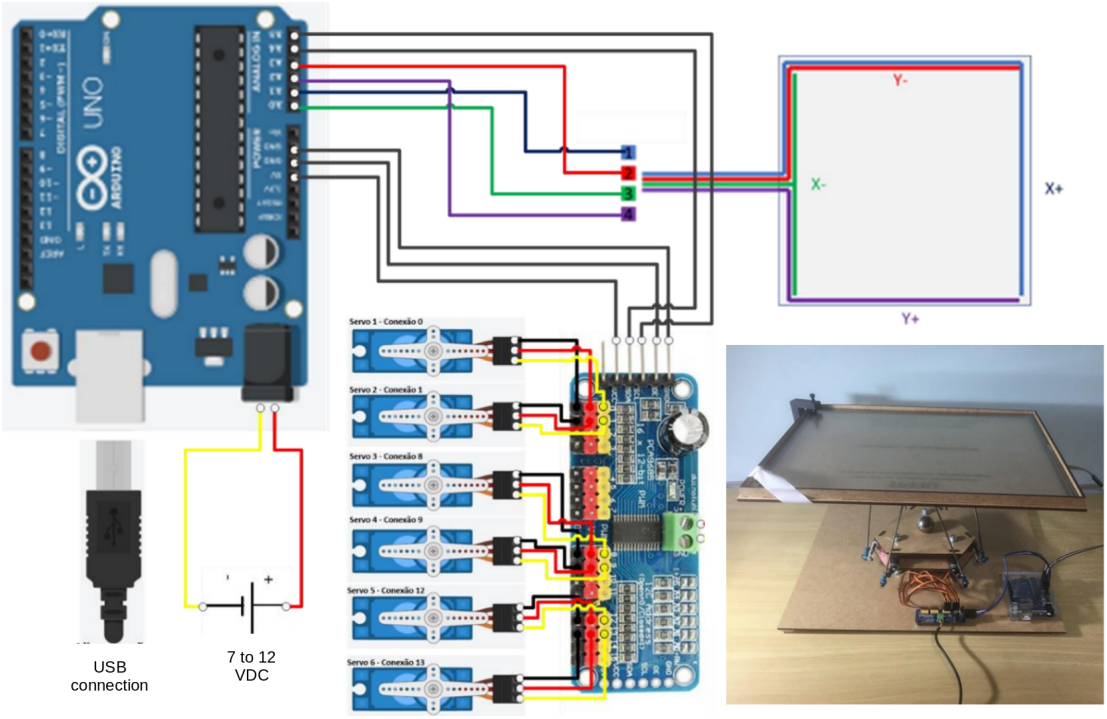


Figure 1. Electronic circuit schematic and a photograph of the physical SGP. Adapted from Vieira [11].

The microcontroller communicates information using the serial output. This signal was used for data acquisition, with the data samples stored external files using the software PuTTY (Release 0.80). These files contained informations about target position, ball positions measured, servo angles and microcontroller timesteps for each datapoint. The microcontroller applied a threshold filter on the sensor measurements, in order to reduce the influence of outliers and spurious measurements. If the measured point at a given cycle was more than 10 mm apart from the previously measured point, it was discarded and a new measurement was taken on the next cycle. This filtering was reflected on the collected data, that already includes this thresholding.

2.1 Mathematical model

Although many mathematical models for the SGP have been proposed, such as in Huang et al. [14], most of them are based on inverse kinematics of the SGP, which starts from a given platform pose and calculates the necessary servo positions. However, the present work deals with a forward kinematics problem, as the system modeled has the servo angles θ_i as inputs, then needs to calculate the platform pose, and finally update the ball position on the plate over time $\mathbf{r}$. This method does not have a closed-form solution, reason why a numerical solver is used to determine the platform pose using constrained optimization. Intermediate joint positions $\mathbf{J}_i$ are computed as:

$$\mathbf{J}_i = \mathbf{B}_i + L_1 \cdot \mathbf{R}(\theta_i, \hat{\mathbf{u}}_i) \cdot \hat{\mathbf{d}}_i. \tag{1}$$

Where $\mathbf{B}_i$ are the base pivot coordinates (mm), L_1 the servo arm length (mm), $\mathbf{R}$ the rotation matrix (from Rodrigues' rotation formula), $\hat{\mathbf{u}}_i$ the precomputed servo rotation axis, and $\hat{\mathbf{d}}_i$ the initial servo arm direction vector. The platform pose is then obtained by minimizing leg-length errors:

$$\min_{\mathbf{p},\mathbf{R}} \sum_{i=1}^{6} \left(\|(\mathbf{p} + \mathbf{R} \cdot \mathbf{P}_i) - \mathbf{J}_i\| - L_2 \right)^2. \tag{2}$$

Where $\mathbf{p} \in R^3$ is the platform center position vector, $\mathbf{R} \in \mathrm{SO}(3)$ the platfrom orientation in space, $\mathbf{P}_i$ the platform leg pivot point, and L_2 the leg length (mm). The ball and plate dynamics are calculated considering the constrained Newton-Euler dynamics with a no-slip condition:

$$\ddot{\mathbf{r}} = \frac{5}{7}\mathbf{g}_{\parallel} - \mu\dot{\mathbf{r}}. \tag{3}$$

$$\omega = \frac{\mathbf{n} \times \dot{\mathbf{r}}}{r}. \tag{4}$$

Where $\mathbf{g}_{\parallel}$ is the gravity component parallel to the plate (mm/s^2), μ the velocity-proportional damping coefficient (s^{-1}), $\mathbf{r}$ the ball position vector, r the ball radius, and $\mathbf{n}$ the plate normal vector. Collisions with the plate walls are handled using an impulse based model with a restitution coefficient ϵ for the velocity normal to the wall v_{normal}:

$$v_{normal}^{new} = -\epsilon \cdot v_{normal}^{old}. \tag{5}$$

2.2 Simulation model

To accurately simulate the SGP-BPS system, a simulation software was developed in Python, using an object-oriented design. Some computational components were developed, those being: a main solver, responsible for connecting the individual modules; a kinematics solver for the Stewart platform; a dynamics solver for the ball and plate; and a visualization system. The order of activation and flow of information between components on a single cycle of the simulation is shown on Fig. 2.

The main solver is responsible to define global parameters, load and store data, ensure the flow of information and call each component. The datasets used for model evaluation contained informations about the individual servo angles (θ_i), ball coordinates and timestamps, each individual dataset providing the development of these values under a single trajectory of the ball on the plate. On average, each file covered a range of 7.4 seconds of operation, with a standard deviation of 3.7 s, for a total of 90 individual experiment runs. Servo angles were passed as inputs for the model, and ball positions were used as the model target when evaluating model error. The dataset had a timestep of around 80 ms, and the simulation timestep (Δt) chosen was of 10 ms. Any deviations of the experimental data from the 80 ms timestep were handled using linear interpolation. Angle inputs over time were updated every 8 simulation timesteps, being kept constant between updates. The full simulation duration was set to match the experimental data length.

The SGP kinematics solver considers mechanical constraints of position and orientation bounds, and also incorporates misalignment compensations due to geometrical imperfections of the physical platform. Each pose update uses the previous result as a starting point for the solver, reducing the number of iterations needed to find the numerical solution. The ball and plate dynamics solver time integration uses an energy conserving Velocity Verlet scheme. Orientation tracking uses a quaternion, avoiding gimbal lock. The visualization framework provides real-time updates of the platform pose and trajectory of the ball. The refresh rate is set to every 10 steps, or 100 ms. Four coordinate plots are used for system monitoring: 3D platform visualization, 2D ball trajectory tracking (both simulated and experimental), polar plate orientation (pitch and roll), and servo angle monitoring over time. The visualizer uses Matplotlib with incremental rendering and path overlays, for cross-platform compatibility.

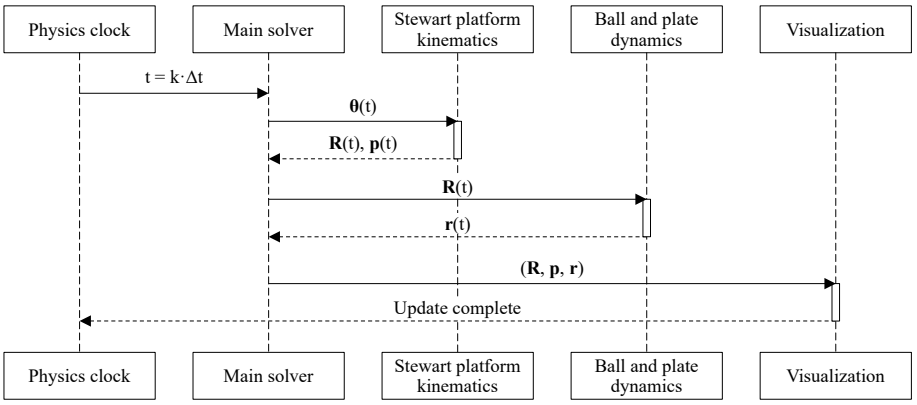


Figure 2. Sequence diagram of the solver developed, for one simulation step k.

3 Results

After running simulations for 90 different datasets of ball trajectories, the simulation outputs were compared to the experimental data through the Euclidean distance (L2 norm) between corresponding trajectory points. This trajectory error values were then evaluated using Mean Absolute Error (MAE), Root Mean Square Error (RMSE), Steady-State Error (SSE) and Maximum Absolute Error (MaxAE) as error metrics. A summary of these metrics is presented on Table 1. Although conveying similar information, MAE and RMSE were both included to calculate the difference between values, making use of the influence of large errors on the RMSE to evaluate the error distribution. SSE calculates the average error over the last 10% of the trajectory, being useful to evaluate final trajectory convergence. Finally, MaxAE provides information about the worst-case deviations in all scenarios.

Table 1. Statistical summary of trajectory error metrics: Mean Absolute Error (MAE), Root Mean Square Error (RMSE), Steady-State Error (SSE) and Maximum Absolute Error (MaxAE). Error values presented in mm.

Metric	count	mean	std	min	25%	50%	75%	max	COV	skewness	kurtosis
MAE (mm)	90	61.9	27.9	13.5	41.9	61.0	78.1	136.6	0.45	0.49	-0.08
RMSE (mm)	90	67.1	29.4	14.2	44.6	65.4	86.4	144.0	0.44	0.47	-0.15
SSE (mm)	90	68.1	35.7	2.7	40.8	64.9	94.6	146.6	0.53	0.26	-0.68
MaxAE (mm)	90	110.2	44.3	20.0	80.8	104.3	132.5	239.0	0.40	0.57	0.39

It can be observed on Table 1 that the mean RMSE (67.1 mm) is 8% larger than the mean MAE, showing also a larger standard deviation. This indicates the presence of large errors in some trajectories, increasing the RMSE due to the squaring of the error values. The SSE had a 10% higher mean and 28% higher standard deviation than MAE, showing high convergence variability on the final portion of the trajectory across experiments. This could mean that the position error tends to increase as the simulation develops over time, diverging incrementally from the experimental values.

Further insights into the error distribution can be obtained by comparing the violin plots in Fig. 3 with the metrics presented in Table 1. All metrics show positive skew, meaning the distribution is more biased towards lower-error experiments. Kurtosis is near zero for MAE and RMSE, configuring a normal-like error distribution. SSE presents significant negative kurtosis, with an error distribution flatter and larger presence of extreme values. This wider error spread shows inconsistency in the final position convergence, when compared to the full trajectory statistics. MaxAE has a longer upper tail and several outliers represented, revealing the presence of occasional severe deviations from the experimental values. However, it's coefficient of variation (COV) is notably smaller than the other metrics, showing that the general error spread is consistent relative to the scale of MaxAE values.

Figure 4 provides us with a visual representation of the experimental and simulated trajectories, for several single experiment datasets. It can be noted that the general shape and behavior of both trajectories shows great similarity. All of them have similar general directions, and a target overshoot followed by corrections that resemble

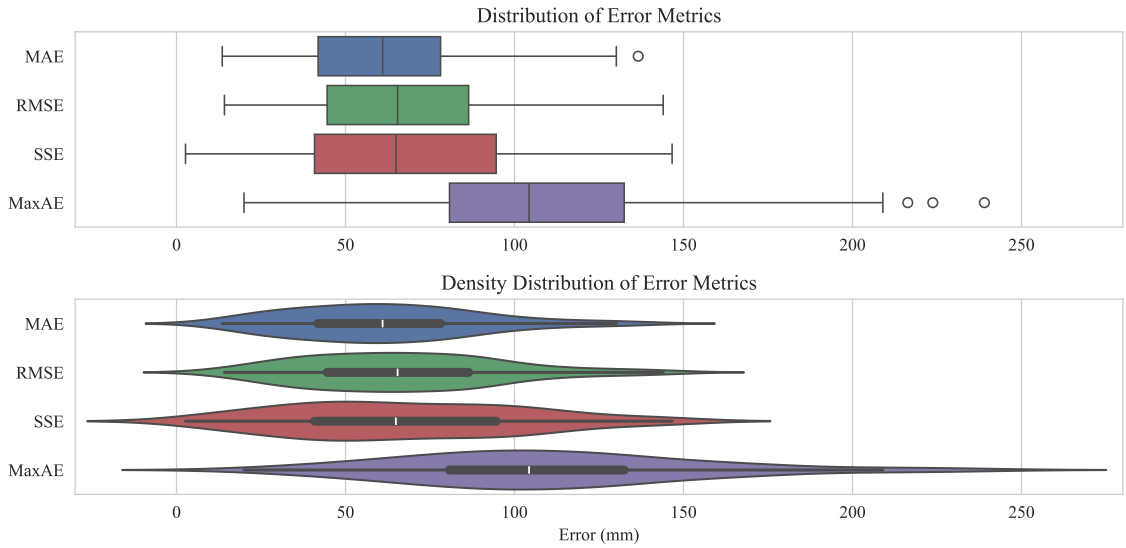


Figure 3. Distribution and density distribution for Mean Absolute Error (MAE), Root Mean Square Error (RMSE), Steady-State Error (SSE) and Maximum Absolute Error (MaxAE).

each other. For example, experiment E9.2A4_5 (top right corner) starts with a leftward trajectory, followed by a target overshoot in the X coordinate that is corrected but induces an upward movement of the ball, also overshooting the target in Y, which then performs a small clockwise loop, finally settling on it's final position. There is, however, an associated SSE, visible in the difference between simulation and experimental final ball positions.

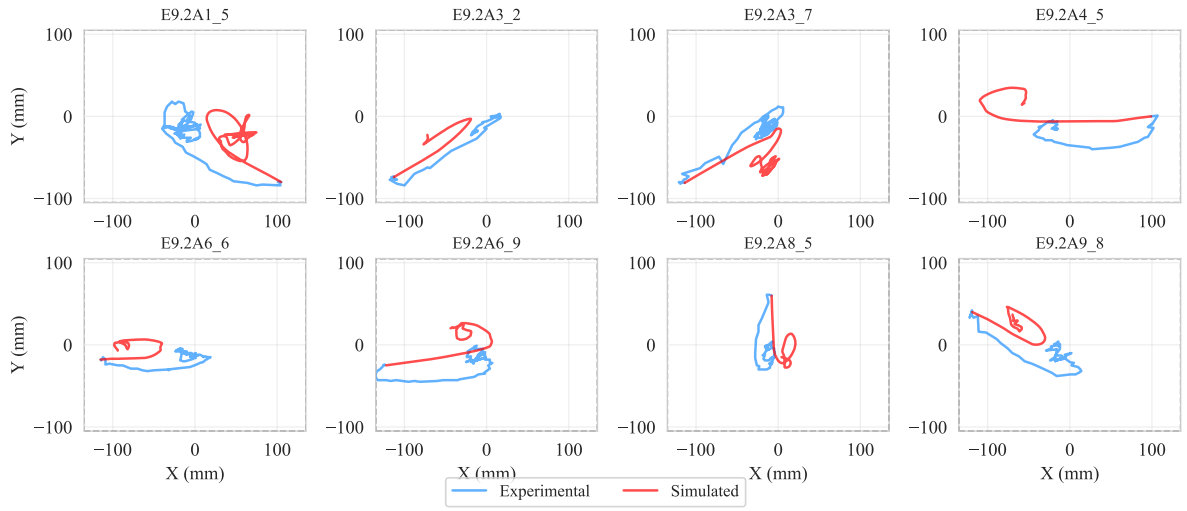


Figure 4. Experimental and simulated trajectories, demonstrating shape similarity between simulated and actual ball tracking. Each pair of trajectories (A1_5,A3_2...) presented has the same starting position for the ball.

The simulated trajectories seem to present overall shape similarity, having however a combination of skewing and translation in relation to the actual trajectories. Some possible causes for these differences could be:

- Initial condition mismatch: for example, differences in the starting velocity between simulated (which considers a static start) and experimental trajectories (that could have an associated start velocity).
- Model parameter errors: incorrect values for rolling resistance, platform dimensions, among others.
- External biases: table tilt, deformations in the mechanical components, wind, vibrations.
- Numerical approximation effects: discretization errors or accumulated numerical drifts, which are inherent to time-integration methods and can be minimized through appropriate step size and solver configuration.
- Systematic model biases: simplifications or assumptions in the model that do not capture real-world asymmetries.

Further development of the analytical model will be performed, aiming to supress or eliminate this skewed condition and reduce overall trajectory error. Some measures that can be experimented are:

- Associate a starting velocity vector to the ball simulation, based on the first n data points of each trajectory.
- Update rolling resistance, table tilt, geometrical parameters of the components.
- Refine the time step size.
- Review model simplifications and evaluate increase in complexity.
- Use data-driven methods to model system imperfections.
- Increase data acquisition rate on the experimental setup to have better discretizations, then run a new set of recorded experiments.

4 Conclusions

An analytical model of the SGP-BPS system was successfully implemented, integrating Stewart platform kinematics (Eqs. 1, 2) and ball dynamics with collision handling (Eqs. 3-5). Validation against 90 experimental trajectories demonstrated strong qualitative agreement in trajectory shapes (Fig. 4) and revealed consistent trends in the error distribution: mean RMSE of 67.1 mm and MaxAE up to 239.0 mm (Table 1). Despite large deviations in specific scenarios, the predominance of low-error cases (Fig. 3) highlights the potential of the model as a reliable tool. Steady-state errors (mean SSE: 68.1 mm) exceeded trajectory-average errors, suggesting error accumulation over time and possible model drift. Primary error sources may include: initial condition mismatches, parameter inaccuracies (rolling resistance, geometry), unmodeled external factors and inherent numerical approximation effects. These findings provide a clear pathway for improvements, and the current model is expected to significantly enhance its predictive accuracy and broaden its applicability in future studies by mitigating the error sources and increasing model precision over time.

Acknowledgements. The authors would like to thank the collaboration of the ANP Human Resources Program for the Petroleum, Natural Gas and Biofuels Sector (PRH-ANP), as well as the Federal University of Paraná (UFPR) and the Laboratory of Monitoring and Control (LaMoC/UFPR) for providing the infrastructure and equipment used in the development of this work. This study was financed, in part, by the São Paulo Research Foundation (FAPESP), Brasil. Process Number 2025/11754-3.

Authorship statement. The authors hereby confirm that they are the sole liable persons responsible for the authorship of this work, and that all material that has been herein included as part of the present paper is either the property (and authorship) of the authors, or has the permission of the owners to be included here.

References

[1] J.-P. Merlet. *Parallel robots*. Number 74 in Solid mechanics and its applications. Kluwer Academic Publishers, Dordrecht ; Boston, MA, 2nd ed edition, 2006.

[2] D. Silva, J. Garrido, and E. Riveiro. Stewart Platform Motion Control Automation with Industrial Resources to Perform Cycloidal and Oceanic Wave Trajectories. *Machines*, vol. 10, n. 8, pp. 711, 2022.

[3] V. E. Gough. Universal tyre test machine. In *Proc. FISITA 9th Int. Technical Congr.*, pp. p. 117–137, London, 1962.

[4] D. Stewart. A Platform with Six Degrees of Freedom. *Proceedings of the Institution of Mechanical Engineers*, vol. 180, n. 1, pp. 371–386, 1965.

[5] J. Xu, Q. Wang, and Q. Lin. Parallel robot with fuzzy neural network sliding mode control. *Advances in Mechanical Engineering*, vol. 10, n. 10, pp. 1687814018801261, 2018.

[6] J. Borras, F. Thomas, and C. Torras. New Geometric Approaches to the Analysis and Design of Stewart–Gough Platforms. *IEEE/ASME Transactions on Mechatronics*, vol. 19, n. 2, pp. 445–455, 2014.

[7] D. Li, S. Wang, X. Song, Z. Zheng, W. Tao, and J. Che. A BP-Neural-Network-Based PID Control Algorithm of Shipborne Stewart Platform for Wave Compensation. *Journal of Marine Science and Engineering*, vol. 12, n. 12, pp. 2160, 2024.

[8] A. Kassem, H. Haddad, and C. Albitar. Commparison Between Different Methods of Control of Ball and Plate System with 6DOF Stewart Platform. *IFAC-PapersOnLine*, vol. 48, n. 11, pp. 47–52, 2015.

[9] R. Kang, Z. Dong, and J. He. Digital twin for mobile robot modeling and simulation. *Journal of Physics: Conference Series*, vol. 2803, n. 1, pp. 012060, 2024.

[10] X. Zhang, Z. Dong, and J. He. Digital twin and hardware-in-the-loop based robotic arm modelling simulation. *Journal of Physics: Conference Series*, vol. 2803, n. 1, pp. 012053, 2024.

[11] V. J. L. Vieira. *Plataforma de Stewart: desenvolvimento de um robô de estrutura paralela e implementação do controle PID*. Undergraduate thesis, Universidade Federal do Paraná, Curitiba, 2022.

[12] W. R. Lüdtke. *Implementação de controle utilizando aprendizado de máquina para um sistema de bola e placa acoplado a uma plataforma Stewart-Gough.* Undergraduate thesis, Universidade Federal do Paraná, Curitiba, 2024.

[13] W. R. Lüdtke and G. S. Venter. Desenvolvimento de um dataset para treinamento de algoritmos de controle não linear baseados em dados utilizando um sistema bola e placa. In *XXX Congresso Nacional de Estudantes de Engenharia Mecânica*. ABCM, 2024.

[14] D. Huang, L. Hui, Q. Lin, B. Bao, and Z. Jin. Modeling and Simulation Analysis of the Stewart Platform System Based on Modelica. In L. C. Jain, V. E. Balas, Q. Wu, and F. Shi, eds, *Frontiers in Artificial Intelligence and Applications*. IOS Press, 2024.