

Article

Dynamic Modeling and Control of a Parallel Mechanism Used in the Propulsion System of a Biomimetic Underwater Vehicle

Juan Antonio Algarín-Pinto ¹, Luis E. Garza-Castañón ^{1,*}, Adriana Vargas-Martínez ¹ and Luis I. Minchala-Ávila ²

¹ School of Engineering and Sciences, Tecnológico de Monterrey, Eugenio Garza Sada 2501, Monterrey 64849, Mexico; a00826540@itesm.mx (J.A.A.-P.); adriana.vargas.mtz@tec.mx (A.V.-M.)

² School of Engineering and Sciences, Tecnológico de Monterrey-Guadalajara, Gral. Ramón Corona 2514, Guadalajara 45138, Mexico; ismael.minchala@tec.mx

* Correspondence: legarza@tec.mx

Featured Application: Achievement of proper thrust for BAUVs based on the correct application of control when parallel mechanisms are incorporated in propulsion systems.

Abstract: Incorporation of parallel mechanisms inside propulsion systems in biomimetic autonomous underwater vehicles (BAUVs) is a novel approach for motion generation. The vehicle to which the studied propulsion system is implemented presents thunniform locomotion, and its thrust depends mainly on the oscillation from its caudal fin. This paper describes the kinematic and dynamic modeling of a 3-DOF spherical 3UCU-1S parallel robotic system to which the caudal fin of a BAUV is attached. Lagrange formalism was employed for inverse dynamic modeling, and its derivation is detailed throughout this paper. Additionally, the implementation of control strategies to compute forces required to actuate limbs to change platform's flapping frequencies was developed. Four controllers: classic PD, a feedforward plus feedback PD, an adaptive Fuzzy-PD, and a feedforward plus Fuzzy-PD were compared in different simulations. Results showed that augmenting oscillating frequencies (from 0.5 to 5 Hz) increased the complexity of the path tracking task, where the classic control strategy (i.e., PD) was not sufficient, reaching percentage errors above 9%. Control strategies using feedforward terms combined with adaptive feedback techniques reduced tracking error below 2% even during the presence of external disturbances.

Keywords: adaptive control; biomimetic autonomous underwater vehicle (BAUV); dynamic modelling; feedforward plus feedback controllers; fuzzy logic; parallel mechanism; propulsion system



Citation: Algarín-Pinto, J.A.; Garza-Castañón, L.E.; Vargas-Martínez, A.; Minchala-Ávila, L.I. Dynamic Modeling and Control of a Parallel Mechanism Used in the Propulsion System of a Biomimetic Underwater Vehicle. *Appl. Sci.* **2021**, *11*, 4909. <https://doi.org/10.3390/app11114909>

Academic Editor: Antoni Grau

Received: 8 April 2021

Accepted: 29 April 2021

Published: 27 May 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Unmanned underwater vehicles, intelligent enough to follow specific objectives and tasks, have been largely developed in recent years. These vehicles have found a large variety of applications in scientific, military and industrial fields [1]. Their development tendencies as well as new design concepts vary from large systems to biomimetic mechanisms [2]. The use of biomimetic autonomous underwater vehicles (BAUVs) represents a great strategy for exploration tasks inside rivers, lakes, and oceans; for underwater inspection; and for ecosystem monitoring. Moreover, what biomimetic robots try to attain is to deliberately replicate a natural system's performance into man-made systems to correctly deliver tasks without disrupting the environment's natural cohesion. Building and designing fish-like systems for underwater vehicles ensures highly efficient motion, hovering and agility of response under water [3]. Just as fish propel their bodies by only using their fins, BAUVs can swim when they are properly equipped by mechanisms that correctly imitate natural propellants. Control over BAUVs' maneuverability will then directly depend on its designed propulsion system. Correct motion inside water assures and increases their chances of doing a job efficiently.

BAUVs travel in water using undulatory or oscillatory propulsion. In this technique, a pressure difference is created in water by moving some part of their body in a wave-like pattern. To generate oscillation in the second-last part of the vehicle's body and/or its caudal fin, different types of mechanisms may be employed. Traditional multi-joint robotic propellers have been designed to properly imitate fish-like swimming by allowing motion on robot's tail in an undulatory fashion. These mechanisms may use simple configurations based on two serially linked motors to generate sideways flapping [4] or dorsoventral flapping [5]. Furthermore, larger configurations can be designed to improve cruising speeds and turning, where three [6] or four motors [7] are linked for such purpose. Some of these multi-joint designs are supplemented with the addition of smart materials such as memory alloy wires to enhance the shaped induced propulsion [8]. By generating rhythmic movements on joints from a serial link of actuators, oscillatory amplitude, frequency, and phase are controlled to manage the flapping from the vehicle's propulsion system [9]. However, these designs present several deficiencies, such as generation of noise, higher energy consumption, and a unidirectional thruster vector [10]. Furthermore, many BAUV swimming capabilities are way less efficient than those found in real fish. In contrast to the novel BAUV-designs, fish can swim at speeds ranging from 3 BLs^{-1} (body length per second) and up to 10 BLs^{-1} , while the oscillation frequency of their bodies and/or caudal fins might get up to 5 Hz depending on the species [11]. Novel solutions to improve control and efficiency in swimming locomotion inside BAUVs are being developed to excel the performance of such traditional propulsion systems [12].

Parallel robotic mechanisms have become a novel approach to generate oscillation inside BAUV's propulsion systems. These systems are made up of one fixed platform and one moving platform linked by several limbs, allowing an end effector to reach many poses by controlling multiple combinations from limbs' dispositions [13]. Hence, end effectors attached to the moving platform can attain wider ranges of workspace. According to [14], limbs used on parallel robots often consist of an actuated prismatic joint connected to the platforms through passive spherical and/or universal joints. The use of linear actuators allows the system to only deal with compression or traction and not bending, which increases position accuracy. Since multiple links move and support the same platform, the use of parallel mechanisms brings higher structural stiffness, a quality desired in most of the cases in propulsion systems. Finally, parallel mechanism designs distribute the mass of the end effector into the number of actuators employed, producing higher stability and better support, enhancing efficiency when moving. The implementation of parallel mechanisms inside propulsion systems on AUVs has been developed in order to attain vectored thrust. This novel approach has caused the reduction of multiple propellers along the AUV's body to only one, to which orientation and position is controlled by means of the incorporation of spherical parallel mechanisms [15], two degrees of freedom (DOF) [16], 3 DOF [17], and even the incorporation of 6 DOF Stewart-Gough platforms [18].

Kinematics and dynamics modelling of parallel robots is a fundamental task to properly control and simulate the system's behavior. This has been traditionally performed through different methodologies such as (and mainly adopted by) the Newton–Euler method, the Lagrange formulation and/or the principle of virtual work. Dasgupta and Choudhury [19] developed the dynamic modelling of a 6 DOF Stewart–Gough platform using the Newton–Euler methodology. Likewise, Bingul and Karahan [20] developed the dynamics model of the same system using the Lagrange approach, while Staicu [21] employed the principle of virtual work. Arian [22] compared methodologies of Newton–Euler and virtual work principle in the dynamics analysis of a 3 DOF Gantry–Tau translational parallel robot. Hraiech et al. [23] developed the reduced inverse dynamic model of a 3 DOF translational platform using the Lagrange formalism. From the latter investigation, the limbs presented a 3 UPU (universal-prismatic-universal) configuration, very similar to the novel design presented by Aparicio et al. [10] for the propulsion system inside a BAUV. The results obtained in the former investigations show that the Lagrange formalism presents

less computational cost and better efficiency and is more adequate when control strategies and simulations are developed.

Control applied to parallel robotic systems ranges from classic theories to new intelligent control approaches. Elkady et al. [24] proposed a multi-simulation analysis for path tracking of their cartesian parallel manipulator with 3 PRRR (prismatic-revolute-revolute-revolute) configuration. In their experiments, the path tracking error was reduced by the implementation of a PD controller and a PD with feedforward controller. Results attained showed that the second control strategy allowed one to achieve the desired trajectory with a position error less than 10^{-5} m after only 0.3 s, which represented an enhancement of almost three times the reduction of error compared to the classic PD. Moreover, adaptive controllers have also been implemented as an alternative for the error's reduction problem when following trajectories. Self-tuning PID controllers have demonstrated the advantages of iteratively adjusting to reduce path tracking errors. The decision-making stage for the gain tuning is commonly implemented by fuzzy logic controllers. Some implementations of adaptive fuzzy-PID controllers applied to parallel manipulators have been developed to properly reach position and orientation from end effectors when working with complex mechanisms. For instance, Wu and Handroos [25] developed a set of 49 fuzzy-rule system to correctly tune a PID controller to position and orient a 6 DOF hydraulic servo control platform, while Khamsehei et al. [26] proposed a set of 81 fuzzy-rule self-tuning PID system to control motion of a 6 DOF Agile Eye parallel mechanism. Both implementations showed that good transient responses and small steady-state errors could be achieved when using adaptive feedback controllers.

This article describes the formalisms behind the derivation of the kinematics and dynamics model of the parallel robot employed in the novel propulsion system proposed by Aparicio et al. [10]. The BAUV's propeller consists of a 3 DOF parallel mechanism coupled to an artificial caudal fin to produce a vectored thruster. The limbs linking both platforms have a 3UCU-1S (3 universal-cylindrical-universal and 1 spherical joint) configuration that allows for the imitation of flapping thunniform locomotion, where more than 90% of the propulsive force comes from the caudal fin oscillation and larger and more efficient thrust is generated compared to other locomotion fashions [12]. The kinematics configuration from each limb is detailed as well. For the inverse dynamic's derivation, Lagrange formalism was employed. For the system's simulation, and to solve path tracking errors, a PD, a feedforward plus feedback PD, an adaptive Fuzzy-PD and a feedforward plus Fuzzy-PD controller were implemented. Simulations for path tracking from limbs' positions were conducted according to the desired platform's orientation. The accuracy of the platform to follow specific trajectories at desired speeds was analyzed when flapping frequencies were set from 0.5 to 5 Hz. Additionally, the system's performance in the presence of simulated external disturbances was studied as well. Results from simulations showed how working at high frequencies increased the complexity of the path tracking task. However, incorporating the dynamical model as a previous step for forecasting the system's behavior complemented by an adaptive feedback stage was the best control strategy when following desired trajectories at preestablished speeds.

2. Materials and Methods

2.1. Propulsion System's Configuration

The novel propulsion system from the studied BAUV employs a 3 DOF rotational platform with 3UCU-1S limb configuration. Compared to other traditional configurations, the platform was designed for oscillatory motion only. With the oscillation from the platform a caudal fin is moved, producing the propulsion and the thunniform locomotion. The total flapping amplitude for this design ranges from $\pm 30^\circ$. Larger flapping amplitudes and higher flapping frequencies result in higher speed and thrust. Figure 1 shows the mechanical design of the propeller and the parallel mechanism that compounds it.

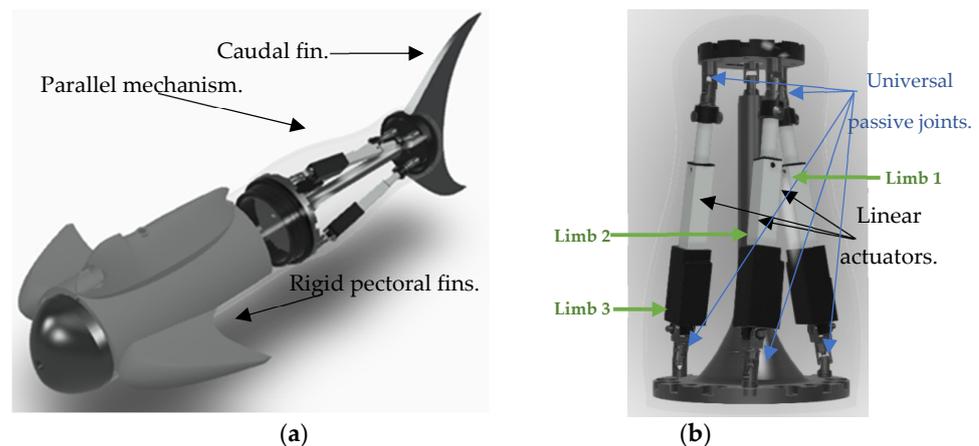


Figure 1. (a) BAUV design with attached caudal fin and body case (b) 3 DOF 3UCU-1S parallel mechanism employed on the propulsion system.

One main feature from this design is its capability to switch between swimming fashions. The moving platform orients the caudal fin according to the desired motion. This allows the vehicle to achieve a vectored thrust and replicate tuna-fish-like or dolphin-like swimming locomotion. From this design, several features should be pointed out:

- The caudal fin is attached to the moving platform. Thrust to the vehicle will be generated by the water displacement produced by the caudal fin. Hence, proper motion control in the platform's oscillation and frequency is of great interest.
- Three limbs are responsible for moving the platform. The parallel mechanism contains three robotic arms with the same configuration. Each limb contains four passive (non-actuated) universal joints at the beginning and at the end where platforms are attached. One linear (prismatic) actuator is placed between such passive joints and is the responsible of the motion generation for each limb.
- The first limb always acts as a pivot. This configuration is helpful when controlling the flapping motion from the platform. This means that the oscillation from the moving platform will be caused by moving only limbs 2 and 3.
- No translational motion is desired. The incorporation of a fourth restrictive limb with spherical joint at its end links the origins of both platforms. This causes distance from both origins to always stay constant.
- The rotational motion of the platform is analyzed based on the Euler angles disposition and measured respect to an inertial frame fixed on the origin of the fixed platform. Considering that only limbs 2 and 3 cause the moving platform's oscillation, only two rotational motions of roll (β) and yaw (γ) are produced. The roll motion is produced over the Y axis and presents a total range from $\pm 30^\circ$. Moreover, the yaw motion is produced over the Z axis and it takes values of 0 for vertical flapping and 90° for horizontal flapping. Finally, it is desired that rotational motion from the platform could reach flapping frequencies ranging from 0.5 to 5 Hz.

2.2. Propulsion System's Kinematics Modeling

Kinematics modeling for the moving platform and the limbs system are computed to correctly understand how the propeller can move. The caudal fin's orientation depends directly on the rotational motion from the platform, and water displacement for thrust generation will depend on its flapping frequency. Since three independent symmetric manipulator arms are responsible of moving the platform, kinematics analysis per limb is detailed as well.

2.2.1. Platform’s Kinematics

Figure 2 describes the free body diagram of the parallel mechanism incorporated inside the propulsion system. The origin of the inertial system $\{O_A\}$ is positioned at the center of the circular fixed platform of radius R_A , while the moving coordinate system $\{O_M\}$ finds its origin at the center of the upper platform of radius R_B . The moving platform is connected to the limbs by universal joints B_i , which are separated from each other at a certain angle Φ_{B_i} . Likewise, the other ends of the limbs A_i are connected to the fixed platform by universal joints with an angular separation of Φ_{A_i} . To avoid singularities when switching from swimming fashions, the mechanical design uniformly distributes attachment points A_i , while B_i joints are distributed following an uneven configuration [10]. Moreover, three linear actuators vary their length at certain distances d_i to create motion on the mechanism. Hence, by varying the limbs lengths, the upper platform is oriented with respect to the center of the base platform.

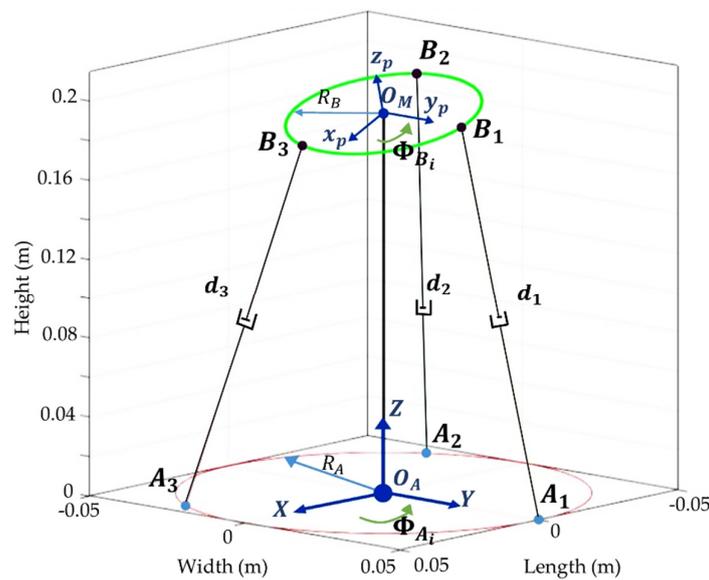


Figure 2. Schematic of the 3 DOF parallel mechanism employed in the propulsion system.

The joint positions $\begin{pmatrix} A \\ B_i \\ P \end{pmatrix}$ in the moving platform are always measured with respect to the base frame $\{O_A\}$. The X, Y, Z coordinate system presents the Z axis pointing upwards and the Y axis pointing to the first universal joint A_1 . In the same way, the x_p, y_p, z_p coordinate system presents the z_p axis normal to the moving platform and the y_p axis pointing to the universal joint B_1 . Since the relative positions $\begin{pmatrix} M \\ B_i \\ P \end{pmatrix}$ of each universal joint with respect to the moving frame are constant for all time, and the position of the origin of the moving frame $\{O_M\}$ with respect to the fixed frame $\{O_A\}$ is always constant as well, it will only be necessary to know the orientation of the moving frame $\{O_M\}$ to compute the position of any universal joint B_i in space. Hence, the location of each universal joint using a homogeneous transformation matrix will be defined by:

$$\begin{bmatrix} A \\ B_i \\ P \end{bmatrix} = \begin{bmatrix} A \\ M \\ R \\ 0_{(1 \times 3)} \end{bmatrix} \begin{bmatrix} A \\ O_M \\ P \\ 1 \end{bmatrix} \begin{bmatrix} M \\ B_i \\ P \\ 1 \end{bmatrix}, \text{ where } i = 1, 2, 3. \tag{1}$$

From Equation (1), $A_M R$ is the rotation matrix from $\{M\}$ with respect to the fixed frame $\{A\}$. In the present analysis, the platform’s orientation is represented by the pitch-roll-yaw Euler angles:

$${}^A_M R = R_Z(\gamma)R_Y(\beta)R_X(\alpha) = \begin{bmatrix} c\beta c\gamma & sas\beta c\gamma - cas\gamma & cas\beta c\gamma + sas\gamma \\ c\beta s\gamma & sas\beta s\gamma + cac\gamma & cas\beta s\gamma - sac\gamma \\ -s\beta & sac\beta & cac\beta \end{bmatrix}, \quad (2)$$

where sa is short notation for $\sin(\alpha)$, $c\beta$ is short notation for $\cos(\beta)$, and so on. Additionally, the position vector ${}^A_{O_M} P$ denotes the constant distance between the origins of both coordinate systems $(P_X, P_Y, P_Z)^T$. Finally, the relative positions ${}^M_{B_i} P$ of each universal joint respect to the moving frame $\{M\}$ are defined as:

$${}^M_{B_1} P = \begin{bmatrix} R_B c(90) \\ R_B s(90) \\ 0 \end{bmatrix}, \quad {}^M_{B_2} P = \begin{bmatrix} R_B c(198) \\ R_B s(198) \\ 0 \end{bmatrix}, \quad {}^M_{B_3} P = \begin{bmatrix} R_B c(342) \\ R_B s(342) \\ 0 \end{bmatrix}. \quad (3)$$

Equations (1) through (3) will be used to determine each universal joint position in space with respect to the fixed coordinate system $\{A\}$. Each limb should be controlled to properly attain those spatial positions. Hence, the analysis of the limbs' configuration and their distribution inside the mechanism should be detailed as well. Table 1 specifies the parameters of the parallel mechanism configuration found in Figure 2.

Table 1. Parameters and disposition of elements in parallel mechanism.

Parameter	Symbol	Value	Unit
Distance between $\{O_A\}$ and $\{O_M\}$.	${}^A_{O_M} P$	$(0, 0, 0.192)^T$	m
Distance from $\{O_A\}$ to joints A_i . ¹	R_A	0.05	m
Distance from $\{O_M\}$ to joints B_i .	R_B	0.025	m
Angular distance between joints A_i .	Φ_{A_i}	$(90, 210, 330)$	deg
Angular distance between joints B_i .	Φ_{B_i}	$(90, 198, 342)$	deg
	α	0	
Roll-pitch-yaw Euler angles.	β	$(-30 \text{ to } 30)$	deg
	γ	$(0 \text{ or } 90)$	

¹ $i = 1, 2, 3$, for all cases.

2.2.2. Limbs' Inverse Kinematics

Each linear actuator d_i should reach spatial positions from each computed universal joint B_i . Based on the schematic in Figure 2, each limb length can be computed according to the distance existing between universal joints $A_i B_i$. By taking the magnitude of such distance, it is possible to know each limb's length while the spatial coordinates of B_i change over time. Hence,

$$d_i = ||\overline{A_i B_i}|| = ||\overline{O_A A_i} + {}^A_{O_M} P + {}^A_M R \cdot {}^M_{B_i} P||. \quad (4)$$

From Equation (4), it is then deduced that desired limbs' lengths will only depend on the desired orientation from the platform and the value of its roll (β) and yaw (γ) motion. Moreover, the kinematics equations of the passive joints could be estimated to facilitate the computation of the reduced inverse dynamics (developed in the next section). Figure 3 shows a closer look at how the rotations are produced inside the designed universal joints. For the further analysis in this article, the joint attaching each linear actuator with its corresponding universal joint is being considered as rigid. Hence, only two rotations ($\theta_{1,i}, \theta_{2,i}$) are considered for the inverse kinematic solutions per universal joint. The solution for each revolute motion will depend directly on the spatial position ${}^A_{B_i} P$ computed to obtain the position of each B_i , where ${}^A_{B_i} P = (B_{ix}, B_{iy}, B_{iz})^T$.

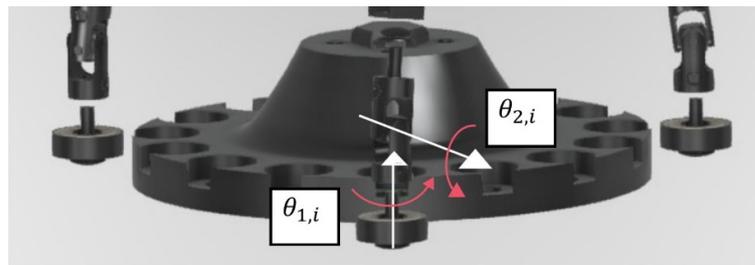


Figure 3. Revolute joints attached on the base of the fixed platform. Two rotations are studied from each universal joint.

A system of six equations will be employed to estimate each passive joint’s position over time, and they are obtained as follows:

For passive joints attaching limb 1, $(\theta_{1,1}, \theta_{2,1})$ inverse kinematics is defined as:

$$\theta_{1,1} = \text{atan2}(B_{1x}, R_A - B_{1y}) \quad (\text{rad}); \tag{5}$$

$$\theta_{2,1} = \frac{\pi}{2} - \text{atan2}\left(B_{1z}, \sqrt{(R_A - B_{1y})^2 + B_{1x}^2}\right) \quad (\text{rad}). \tag{6}$$

where $\text{atan2}(\bullet Y, \bullet X)$ is the two-argument arctangent and 2π variant of $\tan^{-1}\left(\frac{\bullet Y}{\bullet X}\right)$. X and Y subscripts denote the respective components of each ball joint B_i measured from base frame $\{O_A\}$. For passive joints attaching limb 2, $(\theta_{1,2}, \theta_{2,2})$ inverse kinematics is defined as:

$$\theta_{1,2} = \pi + \text{atan2}\left(-\left[R_A \sin\left(\frac{7\pi}{6}\right) - B_{2y}\right], -\left[R_A \cos\left(\frac{7\pi}{6}\right) - B_{2x}\right]\right) \quad (\text{rad}); \tag{7}$$

$$\theta_{2,2} = \frac{\pi}{2} - \text{atan2}\left(B_{2z}, \sqrt{\left(R_A \cos\left(\frac{7\pi}{6}\right) - B_{2x}\right)^2 + \left(R_A \sin\left(\frac{7\pi}{6}\right) - B_{2y}\right)^2}\right) \quad (\text{rad}). \tag{8}$$

Finally, for the passive joints attaching limb 3, $(\theta_{1,3}, \theta_{2,3})$ are computed as:

$$\theta_{1,3} = \text{atan2}\left(-\left[R_A \sin\left(\frac{11\pi}{6}\right) - B_{3y}\right], -\left[R_A \cos\left(\frac{11\pi}{6}\right) - B_{3x}\right]\right) \quad (\text{rad}); \tag{9}$$

$$\theta_{2,3} = \frac{\pi}{2} - \text{atan2}\left(B_{3z}, \sqrt{\left(R_A \cos\left(\frac{11\pi}{6}\right) - B_{3x}\right)^2 + \left(R_A \sin\left(\frac{11\pi}{6}\right) - B_{3y}\right)^2}\right) \quad (\text{rad}). \tag{10}$$

2.3. Propulsion System’s Inverse Dynamic Modeling

The dynamic modeling of parallel manipulators is not straightforward since these systems are based on closed-loop structures. Motion from one limb should be restricted by the other limbs’ motions and the desired pose of the platform. Hence, coupled relationship between system parameters, high nonlinearity in system dynamics and kinematics must be considered [27]. In this article, the reduced inverse dynamic modeling based on the Lagrange formalism was computed. This approach allows one to calculate the actuator forces \mathcal{F}_i while it deals with the passive joints as a set of equations of closed loop chains. With this analysis, forces applied to the line of action on each actuator can be properly computed to create the desired displacement inside the linear actuators. Figure 4 describes the position of the centers of mass from each actuator and the line of action to which force is applied on each limb.

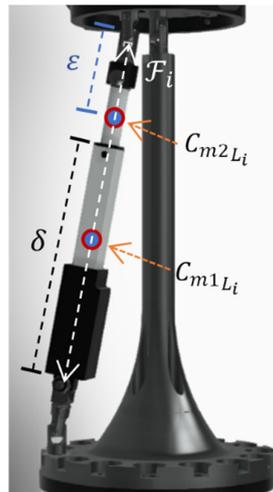


Figure 4. View of an actuated limb and the fourth restrictive limb inside the parallel mechanism; position of centers of mass and force applied to the actuator’s line of action are displayed.

From Figure 4, the parameter δ determines the total length from the actuators’ hull to which center of mass C_{m1Li} is placed at the middle. Moreover, ϵ determines the total distance between the attaching point B_i and the center of mass C_{m2Li} from the moving piston inside the linear actuator. \mathcal{F}_i represents the force applied to the actuator and required to generate motion. Hence, the position of each center of mass (${}^A_{C_{m1Li}}P, {}^A_{C_{m2Li}}P$) inside each limb measured from the inertial frame $\{O_A\}$ can be defined as:

$${}^A_{C_{m1Li}}P = \begin{bmatrix} R_{Ac}(\Phi_{Ai}) + (\delta/2)s_{1i}s_{2i} \\ R_{As}(\Phi_{Ai}) - (\delta/2)c_{1i}s_{2i} \\ (\delta/2)c_{2i} \end{bmatrix}, \quad {}^A_{C_{m2Li}}P = \begin{bmatrix} R_{Ac}(\Phi_{Ai}) + (d_i - \epsilon)s_{1i}s_{2i} \\ R_{As}(\Phi_{Ai}) - (d_i - \epsilon)c_{1i}s_{2i} \\ (d_i - \epsilon)c_{2i} \end{bmatrix} \quad \text{for } i = 1, \quad (11)$$

$${}^A_{C_{m1Li}}P = \begin{bmatrix} R_{Ac}(\Phi_{Ai}) - (\delta/2)c_{1i}s_{2i} \\ R_{As}(\Phi_{Ai}) - (\delta/2)s_{1i}s_{2i} \\ (\delta/2)c_{2i} \end{bmatrix}, \quad {}^A_{C_{m2Li}}P = \begin{bmatrix} R_{Ac}(\Phi_{Ai}) - (d_i - \epsilon)c_{1i}s_{2i} \\ R_{As}(\Phi_{Ai}) - (d_i - \epsilon)s_{1i}s_{2i} \\ (d_i - \epsilon)c_{2i} \end{bmatrix} \quad \text{for } i = 2, \quad (12)$$

$${}^A_{C_{m1Li}}P = \begin{bmatrix} R_{Ac}(\Phi_{Ai}) + (\delta/2)c_{1i}s_{2i} \\ R_{As}(\Phi_{Ai}) + (\delta/2)s_{1i}s_{2i} \\ (\delta/2)c_{2i} \end{bmatrix}, \quad {}^A_{C_{m2Li}}P = \begin{bmatrix} R_{Ac}(\Phi_{Ai}) + (d_i - \epsilon)c_{1i}s_{2i} \\ R_{As}(\Phi_{Ai}) + (d_i - \epsilon)s_{1i}s_{2i} \\ (d_i - \epsilon)c_{2i} \end{bmatrix} \quad \text{for } i = 3. \quad (13)$$

From Equations (11)–(13), $c_{1i}, c_{2i}, s_{1i}, s_{2i}$ are short notations for $\cos(\theta_{1i}), \cos(\theta_{2i}), \sin(\theta_{1i}), \sin(\theta_{2i})$ respectively. Furthermore, the Lagrange formalism is applied from the concepts of work and energy present on the whole mechanism. The Lagrange equation describes the difference between the kinetic and potential energies (E_K, E_P , respectively) from all the moving parts of the system. As previously stated, the parallel mechanism consists of three independent symmetric manipulator arms, whose centers of mass are placed on the same position for each actuator. Additionally, for the present study, it is assumed that the friction forces at all joints are negligible. The total kinetic energy of the whole system is defined as:

$$E_K = E_{K_p} + \sum_{i=1}^3 E_{K_{Li}}, \quad (14)$$

where E_{K_p} is the kinetic energy from the platform and $E_{K_{Li}}$ is the kinetic energy from limb i . Since the platform only presents rotational motion, the equation of kinetic energy E_{K_p} is defined as:

$$E_{K_p} = \frac{1}{2}I_P(\dot{\alpha}^2 + \dot{\beta}^2 + \dot{\gamma}^2), \quad (15)$$

where I_p is the inertial moment computed for the solid disk that compounds the moving platform and $\omega = (\dot{\alpha}, \dot{\beta}, \dot{\gamma})^T$ is the angular velocity of the platform. The center of mass on the moving platform is placed at the origin of the reference frame $\{O_M\}$. On the other hand, the equation of kinetic energy on each limb i will consequently be defined as:

$$E_{K_{Li}} = \frac{1}{2} (\dot{\theta}_{1,i})^2 [m_1 (\delta/2)^2 (s_{2i})^2 + I_L + m_2 (d_i - \varepsilon)^2 (s_{2i})^2 + I_U] + \frac{1}{2} (\dot{\theta}_{2,i})^2 [m_1 (\delta/2)^2 + I_L + m_2 (d_i - \varepsilon)^2 + I_U] + \frac{1}{2} (\dot{d}_i)^2 m_2. \tag{16}$$

In Equation (16), I_L represents the inertial moment from the first part of the limb (actuator's hull) while I_U represents the inertial moment of the upper part of the actuator. Furthermore, the potential energy for the whole system will be defined as:

$$E_P = E_{P_p} + \sum_{i=1}^3 E_{P_{Li}}. \tag{17}$$

Since the parallel mechanism is designed to only follow rotational motions, the potential energy at the moving platform E_{P_p} at its center of mass will not vary over time. It will be defined as:

$$E_{P_p} = M_P g_{O_M^A}^A P; \tag{18}$$

where M_P is the mass of the moving platform, g is the gravitational acceleration and $_{O_M^A}^A P$ is the height presented between platforms. The potential energy at the centers of mass from each limb i is defined as:

$$E_{P_{Li}} = m_1 g [(\delta/2)c_{2i}] + m_2 g [(d_i - \varepsilon)c_{2i}]. \tag{19}$$

Finally, using Equations (14)-(19), the Lagrange equation \mathcal{L} for the whole system is defined as:

$$\mathcal{L} = E_K - E_P. \tag{20}$$

Derivation of Limb's Force Equations

Actuating forces $\mathcal{F}_1, \mathcal{F}_2, \mathcal{F}_3$ applied to the limbs are to be found based on the derivation of the Lagrange equation from the previous analysis. For the present design, the normalized limb lengths d_i are the independent generalized coordinates and the revolute $\theta_{1,i}, \theta_{2,i}$ inside the universal passive joints are the dependent generalized coordinates. Then, the Lagrange equations of motion are defined by:

$$\frac{d}{dt} \left(\frac{\partial \mathcal{L}}{\partial \dot{q}_j} \right) - \left(\frac{\partial \mathcal{L}}{\partial q_j} \right) + \sum_{k=1}^6 \lambda_k \frac{\partial f_k}{\partial q_j} = \mathbb{Q}_j; \quad j = 1, \dots, 9, \tag{21}$$

where:

$$q_j = \begin{cases} d_i & j = i = 1, 2, 3 \\ \theta_{1,i} & j = 4, 5, 6; i = 1, 2, 3 \\ \theta_{2,i} & j = 7, 8, 9; i = 1, 2, 3 \end{cases} \quad \text{and} \quad \mathbb{Q}_j = \begin{cases} \mathcal{F}_i & j = i = 1, 2, 3 \\ 0 & j = 4, \dots, 9 \end{cases}.$$

In real-time applications when controlling these types of mechanisms, the platform's orientation, velocity, and acceleration are known or predetermined. It is based on such predispositions where the interest of determining the forces required to actuate the limbs is of paramount importance. By properly defining the force necessary to actuate the links, the platform will follow a desired trajectory. Equation (21) defines a system of 9 equations to which the first three serve the purpose of computing the values of the forces applied on each actuator (when $j = 1, 2, 3$). Additionally, to correctly solve the system, six constraint equations f_k that are functions of $d_i, \theta_{1,i}, \theta_{2,i}$, and their time derivatives, should be properly

defined. Moreover, the six Lagrange multipliers λ_k should be solved by solving iteratively six simultaneous equations, i.e., Equation (21) when $j = 4, \dots, 9$.

Based on the schematic of the kinematics chains from Figure 2, two vectorial closed loop constraint equations may be defined as follows:

$$\{\vec{O}_A\}A_1 + A_1\vec{B}_1 + B_1\{\vec{O}_M\} + \{\vec{O}_M\}B_2 + B_2\vec{A}_2 + A_2\{\vec{O}_B\} = 0, \quad (22)$$

$$\{\vec{O}_A\}A_2 + A_2\vec{B}_2 + B_2\{\vec{O}_M\} + \{\vec{O}_M\}B_3 + B_3\vec{A}_3 + A_3\{\vec{O}_B\} = 0. \quad (23)$$

Vectorial Equations (22) and (23) are rewritten as scalar equations to generate the six constraint equations f_k where ($k = 1, \dots, 6$):

$$R_{Ac}(\Phi_{A1}) + d_1s_{11}s_{21} + d_2c_{12}s_{22} - R_{Ac}(\Phi_{A2}) - {}^A_M R_{B_1}^M P_x + {}^A_M R_{B_2}^M P_x = 0; \quad (24)$$

$$R_{As}(\Phi_{A1}) - d_1c_{11}s_{21} - {}^A_M R_{B_1}^M P_y + {}^A_M R_{B_2}^M P_y + d_2s_{12}s_{22} - R_{As}(\Phi_{A2}) = 0; \quad (25)$$

$$d_1c_{21} - d_2c_{22} - {}^A_M R_{B_1}^M P_z + {}^A_M R_{B_2}^M P_z = 0; \quad (26)$$

$$R_{Ac}(\Phi_{A2}) - d_2c_{12}s_{22} - {}^A_M R_{B_2}^M P_x + {}^A_M R_{B_3}^M P_x - d_3c_{13}s_{23} - R_{Ac}(\Phi_{A3}) = 0; \quad (27)$$

$$R_{As}(\Phi_{A2}) - d_2s_{12}s_{22} - {}^A_M R_{B_2}^M P_y + {}^A_M R_{B_3}^M P_y - d_3s_{13}s_{23} - R_{As}(\Phi_{A3}) = 0; \quad (28)$$

$$d_2c_{22} - d_3c_{23} - {}^A_M R_{B_2}^M P_z + {}^A_M R_{B_3}^M P_z = 0. \quad (29)$$

Finally, the actuating force along the i th limb is computed by directly solving Equation (21), using the computation of \mathcal{L} and f_k and its derivatives with respect to d_i . The three equations of force required per limb on the designed system are then defined as:

$$\mathcal{F}_1 = \ddot{d}_1 m_2 - (\dot{\theta}_{1,1})^2 m_2 (d_1 - \varepsilon) (s_{21})^2 - (\dot{\theta}_{2,1})^2 m_2 (d_1 - \varepsilon) + m_2 g c_{21} - \lambda_1 s_{11} s_{21} + \lambda_2 c_{11} s_{21} - \lambda_3 c_{21}; \quad (30)$$

$$\begin{aligned} \mathcal{F}_2 = \ddot{d}_2 m_2 - (\dot{\theta}_{1,2})^2 m_2 (d_2 - \varepsilon) (s_{22})^2 - (\dot{\theta}_{2,2})^2 m_2 (d_2 - \varepsilon) + m_2 g c_{22} \\ - \lambda_1 c_{12} s_{22} - \lambda_2 s_{12} s_{22} + \lambda_3 c_{22} + \lambda_4 c_{12} s_{22} + \lambda_5 s_{12} s_{22} - \lambda_6 c_{22}; \end{aligned} \quad (31)$$

$$\mathcal{F}_3 = \ddot{d}_3 m_2 - (\dot{\theta}_{1,3})^2 m_2 (d_3 - \varepsilon) (s_{23})^2 - (\dot{\theta}_{2,3})^2 m_2 (d_3 - \varepsilon) + m_2 g c_{23} + \lambda_4 c_{13} s_{23} + \lambda_5 s_{13} s_{23} + \lambda_6 c_{23}. \quad (32)$$

Equations (30)-(32) might be rearranged so that the robotic system can be defined by a dynamic equation in matrix form. Hence, the forces required to actuate each limb will be defined as:

$$F_i = D\ddot{d}_i + C(d, \theta, \dot{\theta}_i)\theta_{(1,2),i} + G\theta_{2,i} + \lambda_j; \quad (33)$$

where F_i is the (3×1) matrix containing the required forces $(\mathcal{F}_1, \mathcal{F}_2, \mathcal{F}_3)^T$, D is a (3×3) diagonal mass matrix, C is a (3×6) Coriolis and centrifugal forces matrix, G is the (3×1) gravitational forces matrix, and λ_j is the (3×1) Lagrange multipliers and constraints matrix. By making some manipulation over Equation (33), the final plant to which control strategies are being implemented will be defined as:

$$\ddot{d}_i = D^{-1} [F_i - C(d, \theta, \dot{\theta}_i)\theta_{(1,2),i} - G\theta_{2,i} - \lambda_j]. \quad (34)$$

Figure 5 shows the block diagram of the plant and describes how Equation (34) will be iteratively computed on simulations. As explained throughout the last two sections, the plant will require two main inputs: the forces previously computed by a controller and the estimated values from the passive joints. The plant's output will show the real limb's position over time and will be compared to desired limb's length according to the trajectory defined for the moving platform.

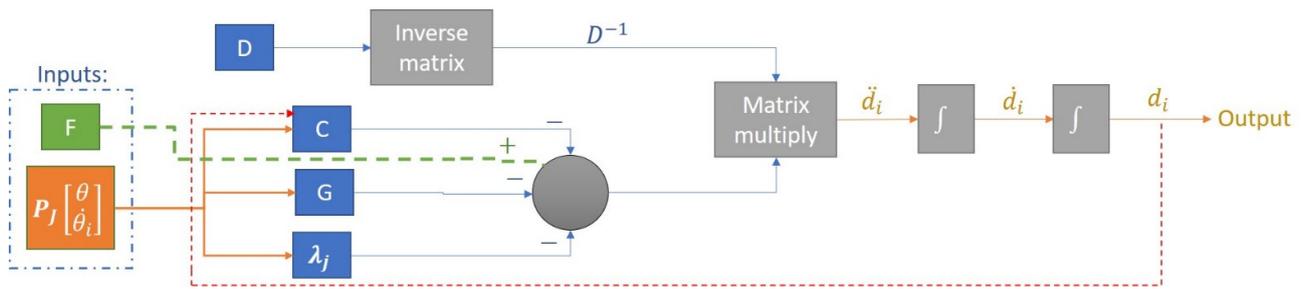


Figure 5. Plant’s block diagram. Forces and passive joints estimated by a controller and the inverse kinematics equation system, respectively, input the plant. Real position of limbs exits the plant.

2.4. Path Tracking and Control

2.4.1. Feedback PD Controller

The moving platform is set to follow a desired trajectory over time. The platform changes its orientation by making a roll motion (β) over the Y axis. As previously stated, the pitch (α) and yaw (γ) motions stay constant through all time. By changing the speed at which the platform changes its orientation, different thrust and water displacement will be produced. Since the caudal fin attached to the moving platform is the only propulsion system for the designed vehicle, attaining a precise control over limbs displacements will be important to achieve the desired motion. Higher flapping frequencies will demand greater control actions and will produce higher thrust levels. Figure 6 describes the block diagram of a closed-loop PD control system where the limb distance d_i is regulated based on the change of the desired platform’s orientation β through time t .

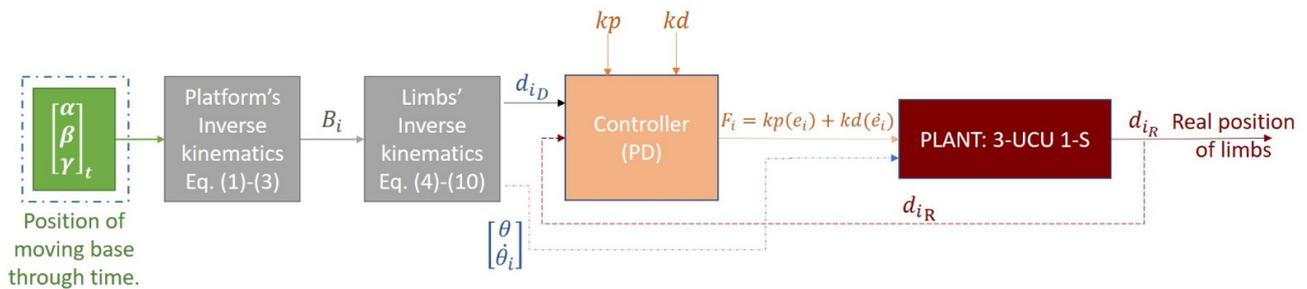


Figure 6. Block diagram for the implementation of the PD controller.

The desired trajectory (orientation) that the platform is set to follow based on pitch-roll-yaw angles through time will be set as:

$$f(\alpha, \beta, \gamma, t) = \begin{cases} \alpha = 0, & t \geq 0 \\ \beta = 30 * \sin(2\pi ft), & t \geq 0 \\ \gamma = 0, & t \geq 0 \end{cases} \quad (35)$$

The flapping frequencies f to which the caudal fin is set to oscillate will range from 0.5 to 5 Hz. The total amplitude of oscillation (by design) is set to $\pm 30^\circ$. Moreover, the control law for the implemented PD is defined as:

$$F_i(t) = kp \cdot e_i(t) + kd \cdot \dot{e}_i(t) \quad , i = 1, 2, 3; \quad (36)$$

where:

$$e_i(t) = d_{iD}(t) - d_{iR}(t), \quad (37)$$

$$\dot{e}_i(t) = \frac{e_i(t) - e_i(t - 1)}{\Delta t} \quad (38)$$

The proportional and derivative gains for the PD controller are arbitrary defined to attain critical damping. Hence, the relation between kp and kd is established by the following relation:

$$kd = 2\sqrt{kp}. \tag{39}$$

2.4.2. Feedforward Plus Feedback Controller

The second control strategy applied intends to attain greater levels of precision when following trajectories. Feedforward controllers consider using the model of the robot's dynamics to proactively generate forces instead of waiting for errors. Inside this strategy, the controller computes the control action based on the desired position, velocity, and acceleration from the trajectory generator, and the system's dynamics [28]. In real applications, platform's dynamic analysis will not always exactly represent real-life circumstances. Moreover, some non-considered disturbances from environment dynamics and inside the mechanism may cause unwanted reactions. Because of this, adding the feedback term (such as the PD controller detailed on the previous section) will deal with all non-considered robot and environment dynamics. Hence, using a feedforward controller based on the dynamic analysis from the studied plant and adding the previous PD term should result in an improved performance for the path tracking problem faced by each limb. Additionally, the proper choice of gains (kp , kd) will ensure exponential decay of the trajectory error. In order to make a fair comparison between these first two control strategies, proportional and derivative gains from the feedback and the feedforward plus feedback controllers will remain constant.

Remembering that (33) presents the dynamic equation for the forces F_i required to actuate each limb, the new control law for the force entering the plant will be reformulated as:

$$f_i = \alpha F'_i + H; \tag{40}$$

where α is the diagonal mass matrix D , H is the sum of the Coriolis, gravitational and Lagrange constraint matrices, and F'_i is the law control from the PD controller. By expanding and solving for Equations (33) and (40), the feedback control output F'_i will be defined as:

$$D\ddot{d}_i + C(d, \theta, \dot{\theta}_i)\theta_{(1,2),i} + G\theta_{2,i} + \lambda_j = DF'_i + C(d, \theta, \dot{\theta}_i)\theta_{(1,2),i} + G\theta_{2,i} + \lambda_j \therefore F'_i = \ddot{d}_i. \tag{41}$$

Moreover, to achieve the linear error dynamics, the desired acceleration on limbs \ddot{d}_{iD} should be computed and the feedback of d_i and \dot{d}_i should be employed to compute position and velocity errors e and \dot{e} . Then, by working on (41), the control action from the PD feedback controller will be defined as:

$$\ddot{d}_i = \ddot{d}_{iD} + kp \cdot e_i(t) + kd \cdot \dot{e}_i(t) \text{ for } i = 1, 2, 3. \tag{42}$$

With the implementation of a feedforward plus feedback controller, it is desired to find a closed loop control where the dynamic behavior of the system is anticipated. This controller should be able to compensate for the dynamics of the system in the presence of perturbations and abrupt changes by working with the accelerations attained inside the whole model. Then, by complementing actions with help of the classic PD controller, error is intended to decay while achieving a smooth performance for the forces applied to the plant. Figure 7 describes the block diagram for the implementation of the feedforward plus feedback controller.

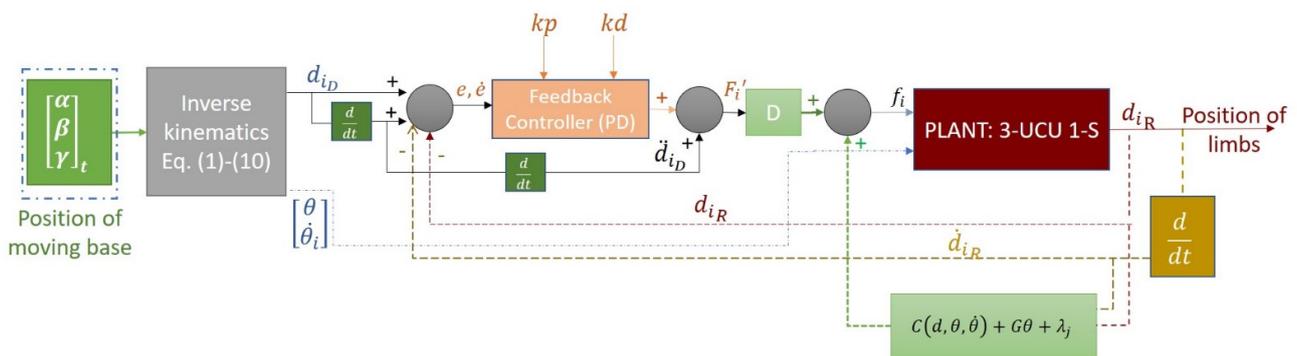


Figure 7. Block diagram for the implementation of the feedforward plus feedback controller.

2.4.3. Adaptive Fuzzy Feedback Controller

When using classic feedback controllers (as a PD) to solve the path tracking problem, the proper tuning of gains will be of great help to accurately reduce error levels when attaining desired trajectories. Furthermore, when using more precise control systems such as a feedforward plus feedback controller, adapting gains could guarantee a better performance and a greater error reduction. This article also details the design of a Fuzzy logic system able to determine which of the gains kp or kd would work better during the feedback phase inside the previous control strategies. Based on the error’s behavior and its time derivative, the fuzzy controller iteratively defines values for the PD’s gains in order to reduce the error rate. Since the goal is to attain high rotation speeds from the moving platform (up to 5 Hz), the control task gets more difficult. At higher flapping frequencies, sudden changes when following preestablished trajectories tend to occur. Hence, an adaptable controller that can adjust based on the change of error will be helpful when computing the limbs acting forces. Figure 8 describes the block diagram of the gain channeling fuzzy logic controller as a previous step for tuning the PD gains inside the feedforward plus feedback control strategy. The Fuzzy logic system for gain tuning was also implemented for the plain feedback controller described in Figure 6.

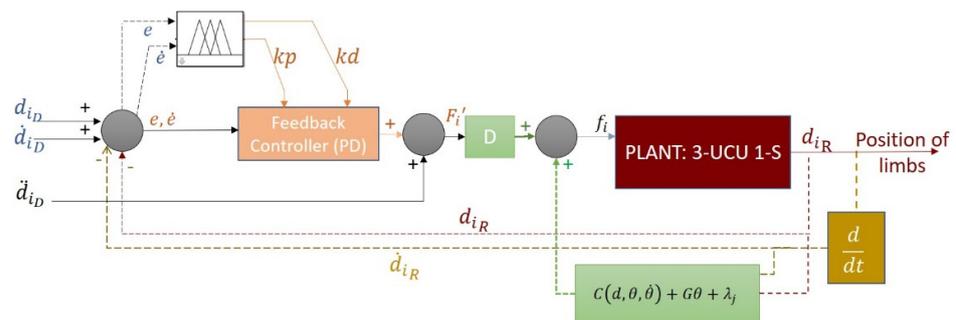


Figure 8. Block diagram of the implementation of the gain channeling fuzzy logic system inside the feedforward plus feedback controller.

Fuzzy logic controllers (FLC) are known for providing human reasoning inside the decision making in the format of rules. The designed FLC is of Mamdani-type. This means that the process of adjusting kp and kd gains will consist of a three-part process: fuzzification, rule-based reasoning, and defuzzification [29]. As seen in Figure 8, the designed Fuzzy-PD controller takes the computations of limbs positional error e and its derivative \dot{e} as inputs and decides to assign coefficients for kp and kd (system’s output) according to a “if-then” set of rules. This means that the PD controller’s gains will be iteratively updated based on the error’s behavior from each limb when following the desired path.

For the fuzzification stage, crisp input values of e and \dot{e} are processed by triangle and trapezoid membership functions (MFs). The linguistic terms for e are large negative (LN),

negative (N), zero (Z), Positive (P) and large positive (LP), where (LN) and (LP) present trapezoid MFs and (N), (Z) and (P) present triangle MFs. On the other hand, only three linguistic terms are defined for \dot{e} : negative (N), zero (Z) and positive (P), where (Z) is a triangle and the other two trapezoid MFs. Two output variables were defined for the defuzzification process: KP and KD. For the defuzzification stage, all linguistic terms for the output variables present gaussian MFs. Even though both output variables are computed separately, they present the same linguistic terms: zero (Z), small (S), medium (M), high (H) and very high (VH). Figure 9 shows the membership functions and the linguistic terms for the inputs and outputs from each FLC. Three FLCs are implemented to compute the kp_i and kd_i , respectively, based on each limb's $e_i(t)$ and $\dot{e}_i(t)$.

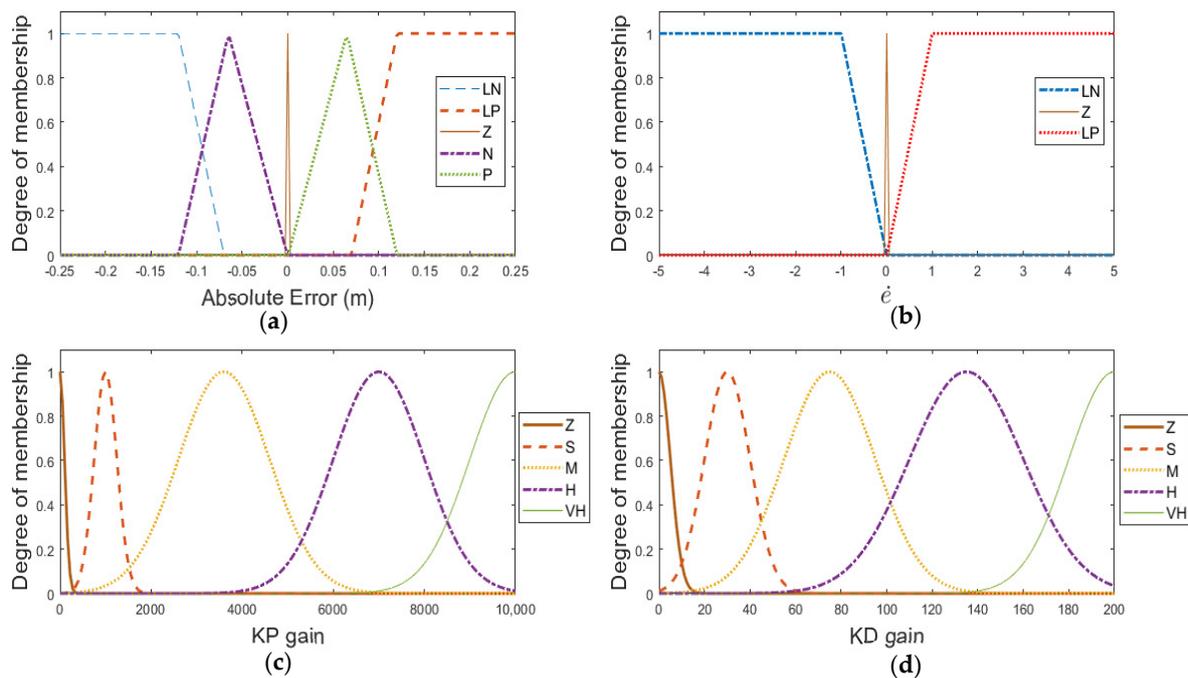


Figure 9. Membership functions for inputs: e_i (a), \dot{e}_i (b), and outputs: kp (c) and kd (d).

Table 2 shows the 15 rules for the inference stage, where inputs' linguistic terms are associated with outputs' linguistic terms based on an if-then structure. The fuzzy rules follow the Mamdani's max-min method, and at the defuzzification stage, the crisp output signal is computed by using the centroid method.

Table 2. Fuzzy control rules for kp_i and kd_i .

kp_i/kd_i		e_i				
		LN	N	Z	P	LP
\dot{e}_i	N	VH/VH	H/H	S/M	H/H	VH/VH
	Z	VH/VH	M/M	Z/Z	M/M	VH/VH
	P	VH/VH	H/H	S/M	H/H	VH/VH

2.5. Simulations

The propulsion system was simulated using Simulink from MatLab®. During simulations, the reduced inverse dynamics developed in the present article were considered as the plant. The platform was set to follow a rotation over the Y axis with roll, pitch and yaw motions defined in (35). From such motions, universal upper joints B_i were computed as well as desired limb lengths. The controlled variable was then the required force \mathcal{F}_i to actuate the limbs so that upper joints might be reached. Four controllers were implemented for such task: PD, Fuzzy-PD (FUPD), Feedforward Plus PD (FFPD), and Feedforward Plus

Fuzzy-PD (FFUPD). Each simulation consisted of the generation of the flapping trajectory over a period of 5 s. For all cases, the flapping total amplitude was set to $\pm 30^\circ$. Moreover, the flapping frequency was increased during simulations from 0.5 to 5 Hz, with steps of 0.5 Hz. The simulations' purpose was of visualizing which controllers could effectively handle the path tracking problem while frequency increased (since high frequencies are desired for thrust generation inside BAUVs). Each controller was responsible for computing the required acting force from each limb separately. Additionally, the same set of fuzzy rules, and input and output variables, were employed for the implementation of the FUPD and FFUPD controllers. Finally, simulations followed the parameters from Table 3.

Table 3. Parameters for the implementation of simulations.

Parameter	Symbol	Value	Unit
Actuator's hull length.	δ	0.068	m
Distance from joint B_i to Cm_{2Li} .	ϵ	0.025	m
Mass of actuator's hull.	m_1	0.054	kg
Mass of actuator's piston.	m_2	0.250	kg
Platform's moment of inertia.	I_p	1.09×10^{-4}	$\text{kg} \cdot \text{m}^2$
Actuator's hull inertial moment.	I_L	3.8×10^{-6}	$\text{kg} \cdot \text{m}^2$
Actuator's piston inertial moment.	I_U	4.5×10^{-6}	$\text{kg} \cdot \text{m}^2$
Gravitational acceleration.	g	9.81	$\text{m} \cdot \text{s}^{-2}$
Proportional PD gain.	kp	2500	-
Derivative PD gain.	kd	100	-

Two set of simulations were conducted:

- Set 1 of simulations: The effectiveness of four controllers was tested based on the comparison of how each limb could reach the desired length when the platform changed its position over time. The precision of following the desired trajectory for each limb was compared for the whole spectrum of flapping frequencies.
- Set 2 of simulations: The desired platform's trajectory and flapping frequencies stayed constant as the previous set, but disturbances were added to each limb's line of action. Three different types of disturbances were simulated and added to the plant: step (S), ramp (R) and sinusoid (Si), mathematically defined by:

$$f(S, t) = \begin{cases} 0, & 0 \leq t < 1; 2 \leq t < 3; t \geq 4 \\ 5, & 1 \leq t < 2; 3 \leq t < 4 \end{cases} (N), \quad (43)$$

$$f(R, t) = \begin{cases} 0, & 0 \leq t < 1; 2 \leq t < 3; t \geq 4; \\ 5(t-1), & 1 \leq t < 2; \\ 5(t-4), & 3 \leq t < 4 \end{cases} (N), \quad (44)$$

$$f(Si, t) = \begin{cases} 0, & 0 \leq t < 1; 2 \leq t < 3; t \geq 4 \\ 5 \sin(17\pi t), & 1 \leq t < 2; 3 \leq t < 4 \end{cases} (N). \quad (45)$$

During each simulation, the percentage relative error $E_{rel(i)}$ was iteratively computed during the steady state for set 1 and during the presence of disturbances for set 2. To better compare the effectiveness from each controller in the path tracking task, the RMS average of $E_{rel(i)}$ was computed at the end of each simulation. Then:

$$E_{rel(i)}(t) = \frac{e_i(t)}{d_{iD}(t)} \times 100 (\%); \quad (46)$$

3. Results and Discussion

Figures 10 and 11 show the results from set 1 of simulations. While Figure 10 describes the path tracking problem for the three limbs at 1 Hz, the latter does it at 5 Hz. Without a

proper control, limbs will fail at attaining desired positions once the flapping frequency starts to increase, which can be visualized by comparing results from both figures. The three limbs could have followed their assigned trajectories when the flapping frequencies were as low as 1 Hz. However, once the flapping frequency of the moving platform increased to 5 Hz, abrupt changes of position could not be correctly overcome by the classic PD controller or the adaptive FUPD controller. The feedforward stage from the FFPD and FFUPD controllers better adapted and reacted to sudden changes and presented better performances while frequency increased. Since limb 1 was assigned as a pivot and did not suffer of abrupt changes, any of the four employed controllers could have served for the task of keeping it in position. However, as seen on Figures 10a and 11a, PD and FUPD present a steady-state error. Since adaptive controllers (FUPD, FFUPD) are iteratively tuning the proportional and derivative gains, they present faster responses when overcoming the initial state error inside all examples. Moreover, it is visualized how limb 2 and 3 move in a synchronized way, hence, for further analysis (and for reasons of brevity) it would only be necessary to focus on the behavior of one of them.

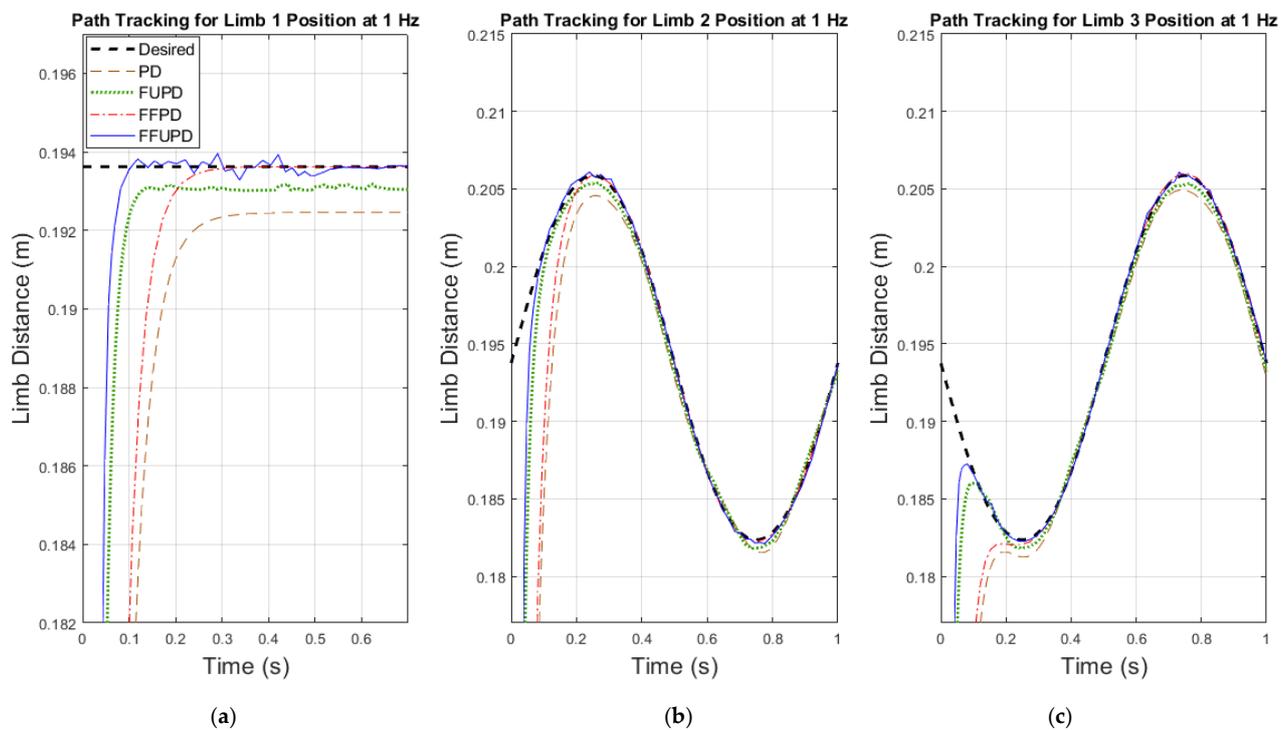


Figure 10. Results for set 1 of simulations at 1 Hz: (a) shows steady-state error from PD and FUPD and great performance to reach position from FFPD and FFUPD, (b) shows great response to follow desired trajectories in limb 2 and (c) limb 3 seems to reach positions with either controller.

Figure 12 presents the control action taken by the adaptive FUPD, FFUPD and the FFPD controllers. The shown graphs from the first row are the required forces for limb 1 to keep position when the platform is flapping at 1 Hz (a through c). The second row of graphs details the required forces to make limb 2 follow a desired trajectory when the platform is flapping at 1 Hz as well (d through f). The final row describes the required forces to move limb 2 when the platform is flapping at 5 Hz (g through h). Since the FUPD controller iteratively adapts according to the error's behavior and its changing rate, shattering is produced on the action of control, as seen on Figure 12b,e,h. The feedforward stage from the FFPD and FFUPD lets the control action be smoother without causing abrupt changes. However, high flapping frequencies will demand high levels of back drive forces applied to limb 2 and 3. FFPD and FFUPD controllers seem to be adequate for such demands without risking precision when reaching desired positions.

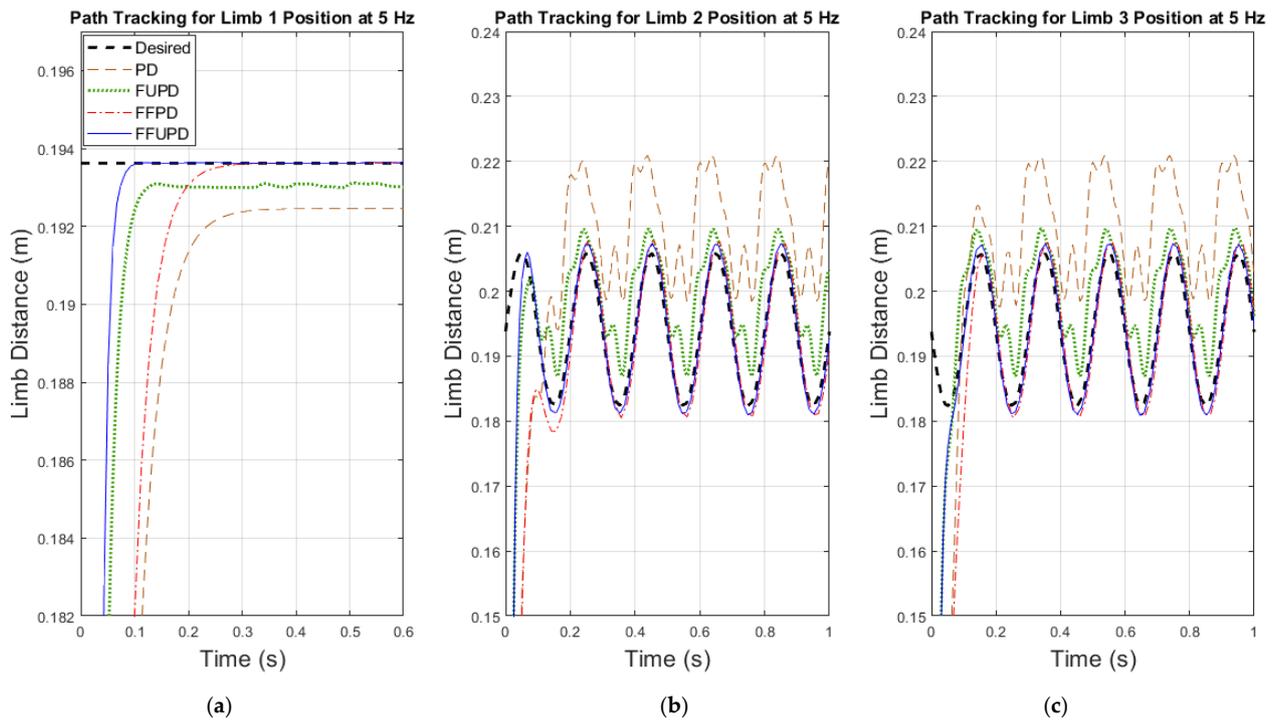


Figure 11. Results for set 1 of simulations at 5 Hz: (a) shows how four controllers do great performance at reaching static positions, however, PD and FUPD never reach desired setpoint; (b,c) show how error increases at high frequencies with PD and FUPD controllers, making limbs 2 and 3 unable to follow desired trajectories.

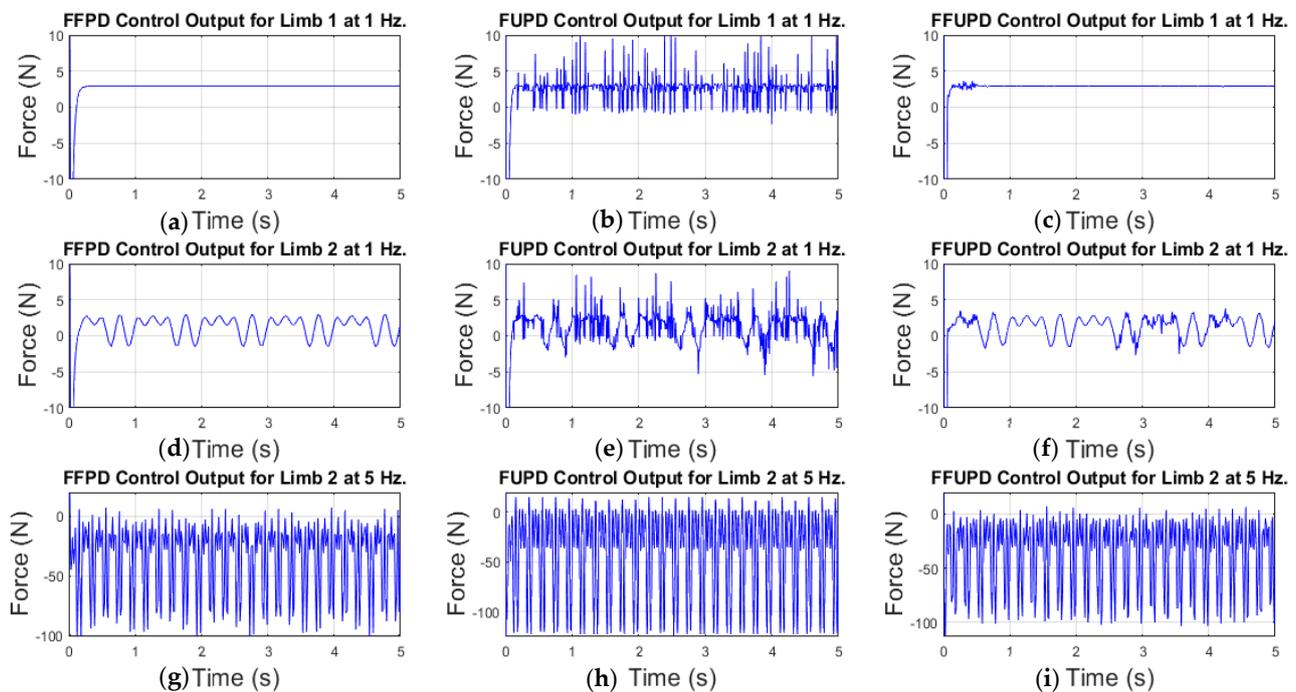


Figure 12. Control action taken from controllers for: (a–c) limb 1 at 1 Hz, (d–f) limb 2 at 1 Hz and (g–i) limb 2 at 5 Hz.

Figure 13 shows the performance of limbs 1 and 2 when the step disturbance (Figure 13a) was applied to their line of action. The comparison between the path tracking error at 1 and 5 Hz is visualized for both limbs. During set 2 of simulations, the three employed controllers were the FUPD, FFPD and FFUPD. Moreover, Figure 14 shows the results attained when the ramp disturbance was applied to limbs 1 and 2, while Figure 15

shows the system's behavior when the sinusoid disturbance was applied as well. In this set of simulations, limbs tend to present a stationary error and an offset once the disturbance is applied. When controlling limb 1, FUPD controller has a better reaction compared to those with a feedforward stage; this is shown in Figure 13b,d, Figure 14b,d and Figure 15b,d. Even when this behavior is also presented for limb 2 and 3, once disturbances are taken away, FUPD seems to never reach the desired set point. FFPD and FFUPD always reach the setpoint when the desired position stays constant through time. Additionally, when the flapping frequencies increase and limbs 2 and 3 has to move faster, FFPD and FFUPD show better results when controlling limbs' lengths even in the presence of disturbances.

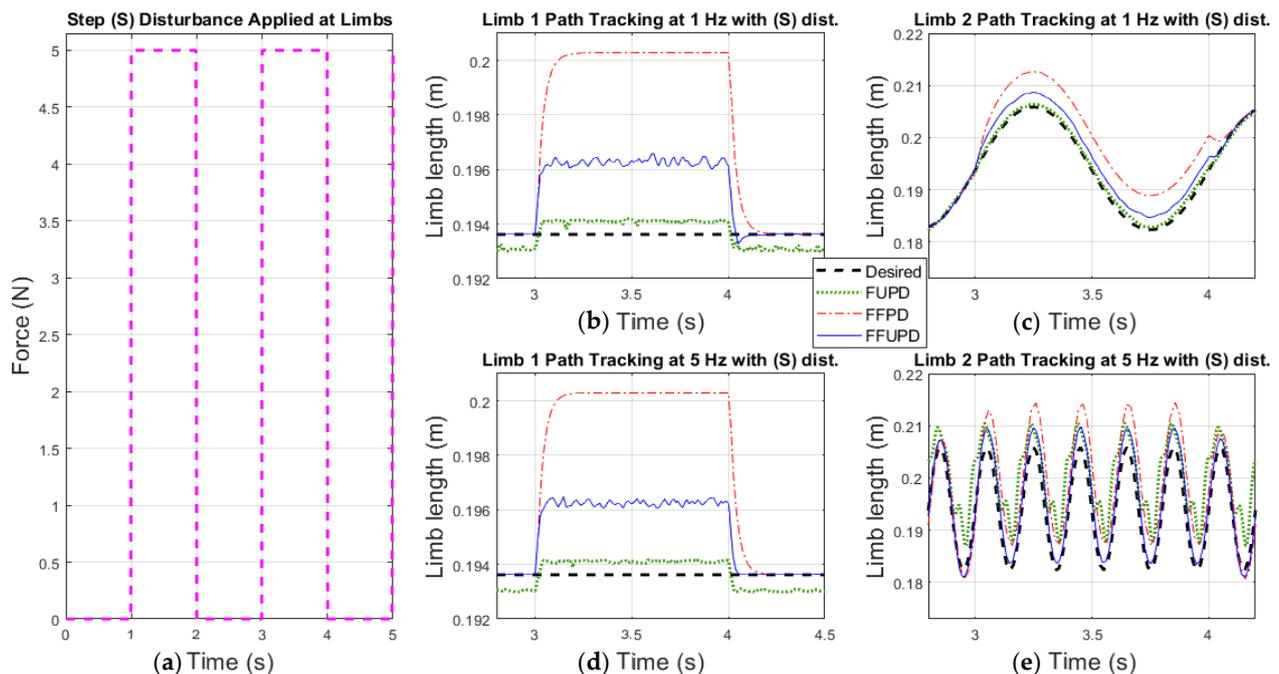


Figure 13. (a) Step disturbance applied to limbs' line of action; (b,d): limb 1 controlled by FUPD, FFPD and FFUPD during perturbation at 1 and 5 Hz respectively; (c,e): limb 2 behavior at 1 and 5 Hz (respectively) during perturbation.

The results achieved show how considering the full dynamics of the system to predict the limb's behavior will serve as a more precise solution when following trajectories than those waiting for error's behavior. Moreover, adapting gains in the feedback stage resulted in a better performance at high frequencies, with the FFUPD being the best choice for controlling the designed system when higher levels of thrust were desired and when disturbances were entering the plant. The way FFUPD better reacts to disturbances applied and allows limbs to follow desired paths is visualized in Figure 13c,e, Figure 14c,e and Figure 15c,e.

Figure 16 shows the controllers' actions when the sinusoid disturbances were applied in limb 2. The comparison was made so that control action could be properly compared when the platform was flapping at 1 and 5 Hz.

Finally, Figure 17 shows the percentage relative error from all simulated tests when limb 2 tried to follow the desired trajectory during a period of 5 s. Flapping frequencies augmented gradually from 0.5 Hz to 5 Hz. As a general conclusion, it is seen that for these types of mechanisms, where higher flapping frequencies are desired, the path tracking problem starts to demand higher and more efficient control strategies. For the designed propulsion system, the platform's oscillation can be properly controlled by the correct application of forces in the limbs that hold it. From results on Figure 17, it is shown that controllers that count with a predictive stage will perform better than those that act according to error rates. Moreover, the adaptive feedforward PD controller presented a stable performance throughout all simulated scenarios. FFUPD controller was able to

predict and adapt when frequencies were increasing and to adapt once the disturbances were applied. FFUPD and FFPD could define limbs' forces so that error would not surpass 2%, which means a reduction of error of about four times the relative error attained by the implementation of the classic PD. Hence, classic feedback controllers such as the designed PD should not be considered to properly handle the path tracking problem at high frequencies.

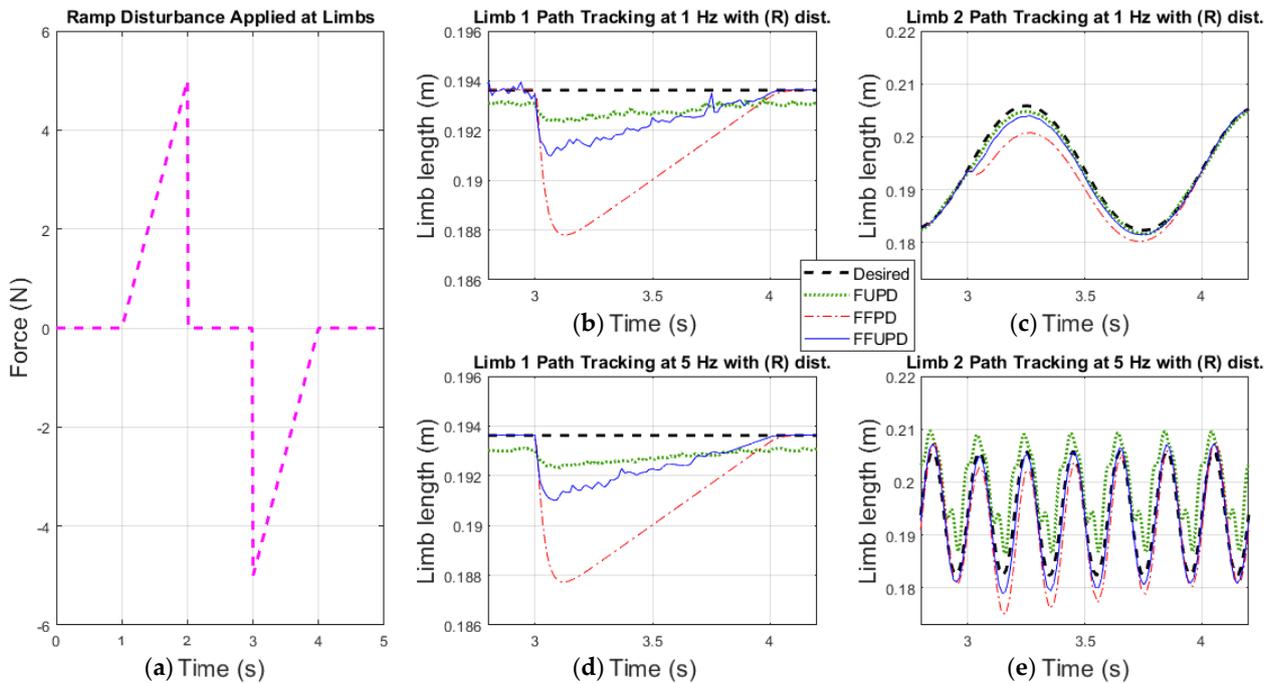


Figure 14. (a) Ramp disturbance applied to limbs' line of action; (b,d): limb 1 controlled by FUPD, FFPD and FFUPD during perturbation at 1 and 5 Hz, respectively; (c,e): limb 2 behavior at 1 and 5 Hz (respectively) during perturbation.

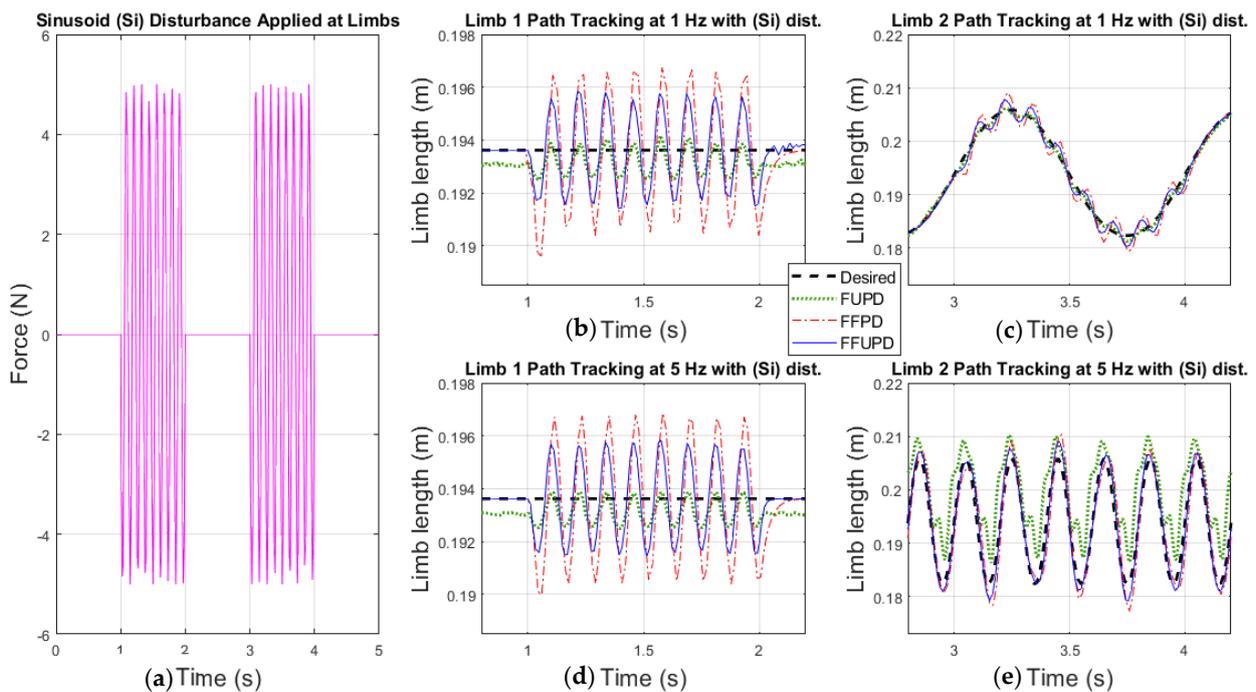


Figure 15. (a) Sinusoid disturbance applied to limbs' line of action; (b,d): limb 1 controlled by FUPD, FFPD and FFUPD during perturbation at 1 and 5 Hz, respectively; (c,e): limb 2 behavior at 1 and 5 Hz (respectively) during perturbation.

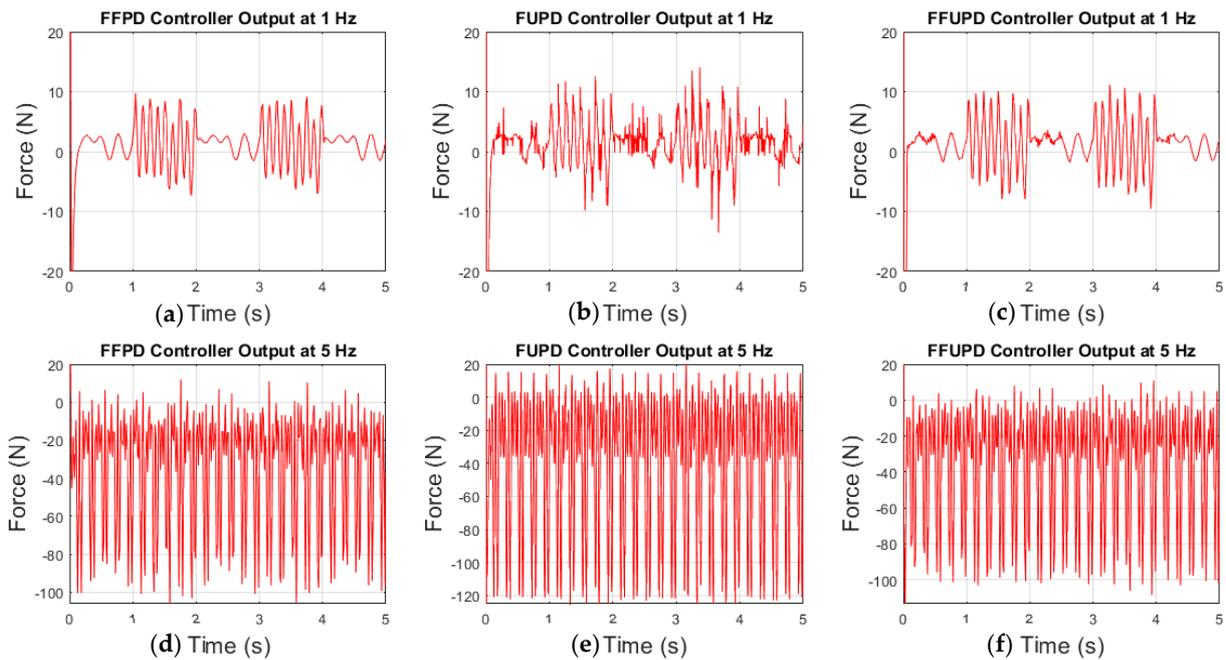


Figure 16. Control action taken by FFPD, FUPD and FFUPD when sinusoid disturbance was applied to limb 2 when: (a–c) platform was flapping at 1 Hz and (d–f) platform was flapping at 5 Hz.

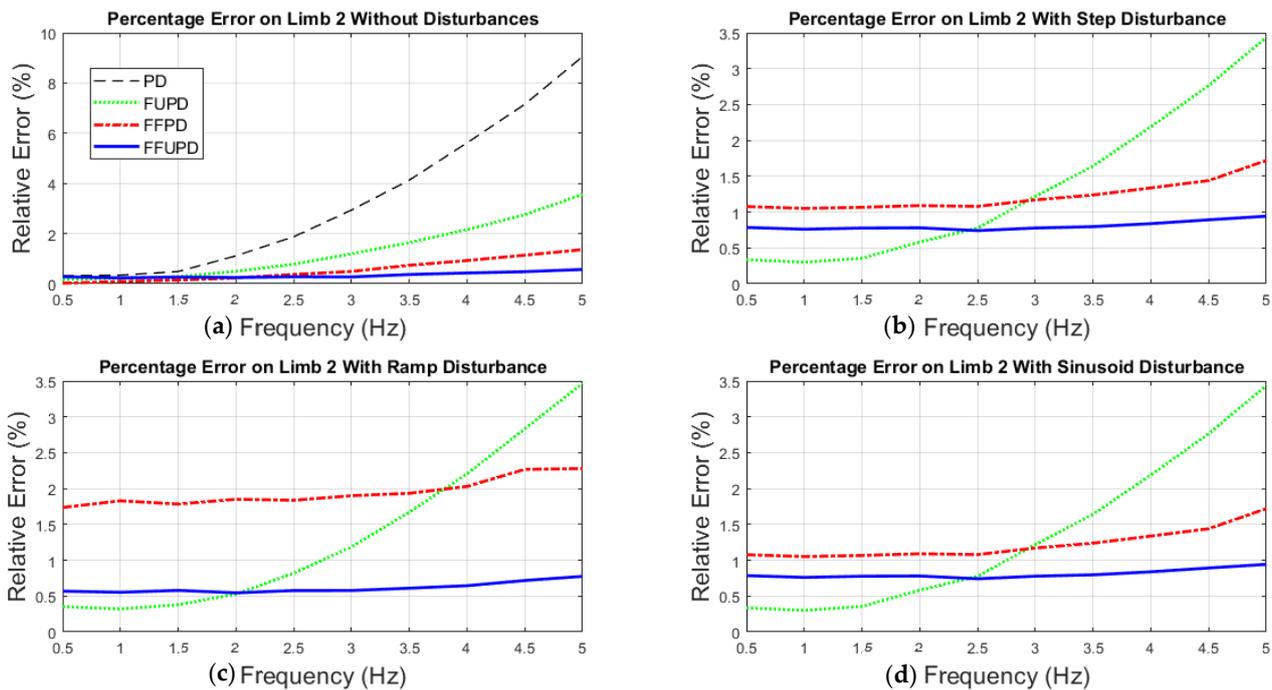


Figure 17. Percentage error in limb 2 when following desired trajectories at different flapping frequencies. Different control strategies employed to reduce error in the absence of external disturbances (a), or in the presence of (b) step, (c) ramp or (d) sinusoid disturbances.

4. Conclusions and Future Work

In this article, the kinematic and inverse dynamic models of a 3-UCU-1S parallel mechanism incorporated inside a novel propulsion system for a BAUV were developed. Since thrust is generated by rotating a caudal fin attached to the moving platform, high levels of flapping frequencies are desired. Simulated cases conducted showed how actuated limbs faced a more complex and challenging path tracking task once the frequencies started

to increase. Four different controllers were designed to solve such a problem, and for all cases, control strategies that considered using the system's dynamics to proactively generate control actions instead of waiting for errors showed more efficient and precise results. Additionally, the incorporation of an adaptive feedback controller as a complement to the feedforward stage showed how the proper choice of gains will ensure a decay of the trajectory error. The fuzzy logic system designed for the PD-tuning task allowed limbs to reduce tracking errors and adapt on to the presence of disturbances.

Further work needs to be done starting with the construction of the designed vehicle based on a flexible silicone-based peduncle to encapsulate the parallel mechanism with the addition of a flexible thermoplastic caudal fin. Physical off-water and underwater experiments should be conducted to understand how these materials will perform when high flapping oscillations (up to 5 Hz) are desired. The control and the path tracking tasks could be further enhanced by the incorporation of more complex techniques for the adaptive control stage. Incorporation of adaptive neuro-fuzzy inference systems and genetic and/or bio-inspired algorithms should be considered as possible solutions for an improved performance from the platform's limbs. Additionally, dynamics simulations for underwater cruising based on the novel propulsion system must be conducted. Path planning strategies based on deep reinforcement learning for a vehicle's trajectory tracking control could be implemented as well.

Author Contributions: Conceptualization, J.A.A.-P. and L.E.G.-C.; methodology, J.A.A.-P. and L.E.G.-C.; software, J.A.A.-P.; validation, L.E.G.-C., A.V.-M. and L.I.M.-Á.; formal analysis, J.A.A.-P. and L.E.G.-C.; investigation, A.V.-M. and L.I.M.-Á.; resources, L.E.G.-C.; data curation, J.A.A.-P., writing—original draft preparation, J.A.A.-P. and L.E.G.-C.; writing—review and editing, A.V.-M. and L.I.M.-Á.; visualization, J.A.A.-P. and L.E.G.-C.; supervision, L.E.G.-C.; project administration, L.E.G.-C.; funding acquisition, A.V.-M. and L.E.G.-C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Tecnológico de Monterrey, Monterrey, NL, Mexico.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author. The data are not publicly available due to privacy.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Sahoo, A.; Dwivedy, S.K.; Robi, P. Advancements in the field of autonomous underwater vehicle. *Ocean. Eng.* **2019**, *181*, 145–160. [[CrossRef](#)]
2. Gafurov, S.A.; Klochkov, E.V. Autonomous Unmanned Underwater Vehicles Development Tendencies. *Procedia Eng.* **2015**, *106*, 141–148. [[CrossRef](#)]
3. Chiu, F.-C.; Guo, J.; Chen, J.-G.; Lin, Y.-H. Dynamic characteristic of a biomimetic underwater vehicle. In Proceedings of the 2002 International Symposium on Underwater Technology (Cat. No.02EX556), Tokyo, Japan, 19 April 2002; pp. 172–177. [[CrossRef](#)]
4. Ren, Q.; Xu, J.; Guo, Z.; Ru, Y. Motion Control of a multi-joint robotic fish based on biomimetic learning. In Proceedings of the 2014 IEEE 23rd International Symposium on Industrial Electronics (ISIE), Istanbul, Turkey, 1–4 June 2014; pp. 1566–1571. [[CrossRef](#)]
5. Yu, J.; Su, Z.; Wu, Z.; Tan, M. Development of a Fast-Swimming Dolphin Robot Capable of Leaping. *IEEE/ASME Trans. Mechatron.* **2016**, *21*, 2307–2316. [[CrossRef](#)]
6. Anderson, J.M.; Chhabra, N.K. Maneuvering and Stability Performance of a Robotic Tuna. *Integr. Comp. Biol.* **2002**, *42*, 118–126. [[CrossRef](#)] [[PubMed](#)]
7. Yu, J.; Tan, M.; Wang, S.; Chen, E. Development of a Biomimetic Robotic Fish and Its Control Algorithm. *IEEE Trans. Syst. Man Cybern. Part B Cybern.* **2004**, *34*, 1798–1810. [[CrossRef](#)] [[PubMed](#)]
8. Suleman, A.; Crawford, C. Studies on Hydrodynamic Propulsion of a Biomimetic Tuna. In *Underwater Vehicles*; IntechOpen: Vienna, Austria, 2008; pp. 459–486. [[CrossRef](#)]
9. Yu, J.; Tan, M. Design and Control of a Multi-joint Robotic Fish. In *Springer Tracts in Mechanical Engineering*; Springer Robot Fish: Berlin/Heidelberg, Germany, 2015; pp. 93–117.

10. Aparicio-García, C.T.; Duchi, E.A.N.; Garza-Castañón, L.E.; Vargas-Martínez, A.; Martínez-López, J.I.; Minchala-Ávila, L.I. Design, Construction, and Modeling of a BAUV with Propulsion System Based on a Parallel Mechanism for the Caudal Fin. *Appl. Sci.* **2020**, *10*, 2426. [[CrossRef](#)]
11. Alvarado, P.V.Y.; Youcef-Toumi, K. Design of Machines With Compliant Bodies for Biomimetic Locomotion in Liquid Environments. *J. Dyn. Syst. Meas. Control.* **2005**, *128*, 3–13. [[CrossRef](#)]
12. Scaradozzi, D.; Palmieri, G.; Costa, D.; Pinelli, A. BCF swimming locomotion for autonomous underwater robots: A review and a novel solution to improve control and efficiency. *Ocean. Eng.* **2017**, *130*, 437–453. [[CrossRef](#)]
13. Taghirad, H.D. *Parallel Robots: Mechanics and Control*; CRC Press: Boca Raton, FL, USA, 2013; Volume 128.
14. Merlet, J.P. *Parallel Robots*; Springer-Verlag: Berlin/Heidelberg, Germany, 2006; Volume 128.
15. Cavallo, E.; Michelini, R.C. A Robotic Equipment for the Guidance of a Vectored Thruster AUV. In Proceedings of the 35th International Symposium on Robotics ISR, Paris, France, 23–26 March 2004; pp. 23–26.
16. Wang, R.; Guo, X.; Zhong, S. An Underwater Vector Propulsion Device Based on the RS+2PRS Parallel Mechanism and Its Attitude Control Algorithm. *Appl. Sci.* **2019**, *9*, 5210. [[CrossRef](#)]
17. Liu, T.; Hu, Y.; Xu, H.; Zhang, Z.; Li, H. Investigation of the vectored thruster AUVs based on 3SPS-S parallel manipulator. *Appl. Ocean. Res.* **2019**, *85*, 151–161. [[CrossRef](#)]
18. Pazmiño, R.S.; Cena, C.E.G.; Arocha, C.A.; Santonja, R.A. Experiences and results from designing and developing a 6 DoF underwater parallel robot. *Robot. Auton. Syst.* **2011**, *59*, 101–112. [[CrossRef](#)]
19. Dasgupta, B.; Choudhury, P. A general strategy based on the Newton–Euler approach for the dynamic formulation of parallel manipulators. *Mech. Mach. Theory* **1999**, *34*, 801–824. [[CrossRef](#)]
20. Bingul, Z.; Karahan, O. Dynamic Modeling and Simulation of Stewart Platform. In *Serial and Parallel Robot Manipulators–Kinematics, Dynamics, Control and Optimization*; Kucuk, S., Ed.; IntechOpen: London, UK, 2012; pp. 19–42. [[CrossRef](#)]
21. Staicu, S. Dynamics of the 6-6 Stewart parallel manipulator. *Robot. Comput. Manuf.* **2011**, *27*, 212–220. [[CrossRef](#)]
22. Arian, A.; Danaei, B.; Abdi, H.; Nahavandi, S. Kinematic and dynamic analysis of the Gantry-Tau, a 3-DoF translational parallel manipulator. *Appl. Math. Model.* **2017**, *51*, 217–231. [[CrossRef](#)]
23. Chouchane, M.; Fakhfakh, T.; Daly, H.B.; Aifaoui, N.; Chaari, F. Design and Modeling of Mechanical Systems-II. In Proceedings of the 10th World Congress on Engineering Asset Management (WCEAM 2015), Hammamet, Tunisia, 23–25 March 2015; Volume 789, pp. 479–487.
24. Elkady, A.; Elkobrosy, G.; Hanna, S.N.; Sobh, S.H.A.T. Cartesian Parallel Manipulator Modeling, Control and Simulation. In *Parallel Manipulators, towards New Applications*; I-Tech Education and Publishing: Vienna, Austria, 2008; p. 506.
25. Wu, H.; Handroos, H. Hybrid fuzzy self-tuning PID controller for a parallel manipulator. In Proceedings of the Fifth World Congress on Intelligent Control and Automation (IEEE Cat. No.04EX788), Hangzhou, China, 15–19 June 2004; Volume 3, pp. 2545–2549.
26. Fadaei, M.H.K.; Zalaghi, A.; Atigh, S.G.R.A.G.; Torkani, Z. Design of PID and Fuzzy-PID Controllers for Agile Eye Spherical Parallel Manipulator. In Proceedings of the 2019 5th Conference on Knowledge Based Engineering and Innovation (KBEL), Tehran, Iran, 28 February–1 March 2019; pp. 113–117. [[CrossRef](#)]
27. Thanh, T.D.; Kotlarski, J.; Heimann, B.; Ortmaier, T. On the inverse dynamics problem of general parallel robots. In Proceedings of the 2009 IEEE International Conference on Mechatronics, Malaga, Spain, 14–17 April 2009; pp. 1–6. [[CrossRef](#)]
28. Lynch, K.M.; Park, F.C. *Modern Robotics: Mechanics, Planning, and Control*; Cambridge University Press: Cambridge, UK, 2017; 544p.
29. Lotfi, M.; Menhaj, M.B.; Hosseini, S.A.; Shirani, A.S. A design of switching supervisory control based on fuzzy-PID controllers for VVER-1000 pressurizer system with RELAP5 and MATLAB coupling. *Ann. Nucl. Energy* **2020**, *147*, 107625. [[CrossRef](#)]