Article

Design of a Gough–Stewart Platform Based on Visual Servoing Controller

Minglei Zhu, Cong Huang, Shijie Song and Dawei Gong *

School of Mechanical and Electrical Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China; minglei.zhu@uestc.edu.cn (M.Z.); 20121040210@std.uestc.edu.cn (C.H.); 201721080102@std.uestc.edu.cn (S.S.)
* Correspondence: pzhzhx@uestc.edu.cn

Abstract: Designing a robot with the best accuracy is always an attractive research direction in the robotics community. In order to create a Gough–Stewart platform with guaranteed accuracy performance for a dedicated controller, this paper describes a novel advanced optimal design methodology: control-based design methodology. This advanced optimal design method considers the controller positioning accuracy in the design process for getting the optimal geometric parameters of the robot. In this paper, three types of visual servoing controllers are applied to control the motions of the Gough–Stewart platform: leg-direction-based visual servoing, line-based visual servoing, and image moment visual servoing. Depending on these controllers, the positioning error models considering the camera observation error together with the controller singularities are analyzed. In the next step, the optimization problems are formulated in order to get the optimal geometric parameters of the robot and the placement of the camera for the Gough–Stewart platform for each type of controller. Then, we perform co-simulations on the three optimized Gough–Stewart platforms in order to test the positioning accuracy and the robustness with respect to the manufacturing errors. It turns out that the optimal control-based design methodology helps get both the optimum design parameters of the robot and the performance of the controller (robot + dedicated controller).

Keywords: parallel robot; image-based visual servoing; optimal design; Gough–Stewart platform; controller singularity; hidden robot

1. Introduction

Parallel robots are becoming more and more attractive due to their better performances compared with classical serial robots in terms of high speed and acceleration, payload, stiffness, and accuracy [1]. Nevertheless, the traditional control of parallel robots is always troublesome because of the high non-linear input/output relations.

It can be found in [2] that a large number of researches have focused on the control of parallel robots. Generally, the only way to ensure high accuracy of a parallel robot is to get the robot model as detailed as possible for the model-based controller [3]. However, due to several factors such as errors from manufacturing and robot assembly, even detailed models still suffer from the problem of inaccuracy in practice. Therefore, more and more researches are currently focusing on finding an alternative controller to sidestep the complex kinematic architecture of the robot and to reach a better positioning accuracy performance compared with the classical model-based controllers. The sensor-based controller is an efficient method that estimates the pose of the end-effector with external sensors [4–6]. Visual servoing is a sensor-based controller, which takes one or several cameras as external sensors and closes the control loop by the vision information obtained from the camera. Visual servoing can be classified into two main groups: position-based visual servoing (PBVS) and image-based visual servoing (IBVS). PBVS directly controls the pose of the target with respect to the camera in Cartesian space [7]. Image-based visual servoing aims
at minimizing the errors between current image features and desired image features directly in image space. It is more robust with the calibration errors compared with PBVS and can make sure that the target is always in the image plane so that we do not lose track when servoing. Therefore, we propose to apply IBVS as the external sensor-based controller in this paper. A large number of researches focused on controlling parallel robots with IBVS with the development of the image processing and image acquisition technology [5,8–14]. It has been proven that the end-effector pose can be estimated effectively throughout the direct observation by vision [15] or the indirect observation [14,16–18]. In addition, the choices of image features applied in visual servoing of parallel robots are numerous, such as image moments [19,20] when the camera can observe the end-effector directly or the observation of robot legs when directly observing the end-effector is difficult to realize (such as the machine tool) [9].

When the vision-based controller is applied to control parallel robots, the positioning accuracy is one of the most important internal performances and the positioning accuracy comes from the error of observation of the image features [21]. The types and numbers of cameras that are used, together with the kinds of image features, all have an influence on the observation error [22]. In addition, the geometric parameters of robots and the camera position also affect the positioning accuracy since they change the interaction models, which leads to effects on the positioning accuracy [23,24]. One problem that should be mentioned is that the mapping between the image feature space and the Cartesian space is not free of singularities [25]. The existence of the singularity of the interaction model has a great influence on the accuracy performance of the parallel robot [26]. In conclusion, in order to ensure the best accuracy performance for the pair {robot +controller} throughout its workspace, the robot geometric parameters and camera position should be optimized in advance.

The optimal design methodology of the robot aims at getting the optimal design geometric parameters of the robot to minimize a given objective under constraints. In [27], when visual servoing is applied to the control of parallel robots, the controller singularity and the internal performance (especially the positioning accuracy) should be taken into account in advance. In addition, the visual servoing controller is never considered in the optimal design process before. Therefore, in this work, the “control-based design” methodology considering the controller performance is developed and the positioning accuracy, together with the controller singularity of the corresponding controllers will be taken into account during the robot design process in order to get the optimal geometric parameters of a Gough–Stewart platform for a dedicated controller with the best performance of accuracy and to avoid the instability issues that appear in the control process. In this case, three types of vision-based controllers will be considered:

- Leg-direction-based visual servoing (LegBVS) [20];
- Line-based visual servoing (LineBVS) [28];
- Image-moment-based visual servoing (IMVS) of a feature mounted on the platform [19].

To the best of our knowledge, this is the first time that we design a spatial 6 DOF parallel robot with the optimal control-based design methodology. In addition, this is the first time that the topological optimization is applied in the image moment visual servoing controller design.

This paper is organized as follows: Section 2 presents the robot architecture, design requirements, and the specifications of visual servoing controllers. The concept of visual servoing applied for controlling the Gough–Stewart platform is reviewed in Section 3. In Section 4, the controller accuracy performance (the error model relating the error from the camera observation to the positioning error of the robot) and controller singularities, which lead to the instability of the robot, are discussed. Optimal design procedure based on the visual servoing controllers is introduced and solved in Section 5. Then, in Section 6, the co-simulation between Simulink and ADAMS with the result analysis are described. Finally, some conclusions are drawn in Section 7.
2. Robot Architecture and Specification

In this paper, we optimize the geometry of the Gough–Stewart platform with visual servoing in order to get excellent performance of the pair robot+controller. The Gough–Stewart platform, also called a hexapod, is a parallel robot with 6 degrees of freedom (DOF): the moving platform of the Gough–Stewart platform translates along the three axes of the space and rotates around the three axes of the space with respect to the fixed base [29]. The Gough–Stewart platform designed in this chapter is a 6-UPS robot (Figure 1a). The moving platform of the robot is linked to the fixed base by 6 individual chains \( B_iP_i \) \((i = 1 \cdots 6)\). The connection of the chains with the base is a U joint located at \( B_i \) \((i = 1 \cdots 6)\), the chains are attached to the end-effector by a S joint located at \( P_i \) \((i = 1 \cdots 6)\) and the prismatic actuator allows the change of the lengths of the links \( B_iP_i \) \((i = 1 \cdots 6)\) (Figure 1b).

The base and the moving platform of the considered Gough–Stewart platform are symmetric hexagons (Figure 1c). The radius of the circumcircle of the base is \( r_b \), and the radius of the circumcircle of the moving platform is \( r_a \). The angle \( \angle B_1B_cB_2 = 2\alpha_1 \), the angle \( \angle P_1P_cP_2 = 2\alpha_2 \) and the angle \( \angle x'P_cP_0 = \alpha_0 \) (Figure 1c).

The complete workspace of the Gough–Stewart platform is a six-dimensional space. We should consider both its 3D location and the orientation of the moving platform. The definition of its orientation workspace is based on the Tilt and Torsion (T&T) angles proposed in [30]. The tilt and torsion angles are defined in two stages: a tilt and a torsion,
In the first stage, as illustrated in Figure 1d, the frame $i$ first rotates about the base $z_i$-axis by an angle $\phi$, then about the $y_i$-axis by an angle $\theta$, then about the $z_i$-axis by an angle $-\phi$, and finally about the new $z_k$-axis by an angle $\sigma$. The expression of the rotation matrix of the T&T angles can be found in [30]. With the T&T angles, a novel 3D workspace subset named maximum tilt workspace was developed in [31]. This workspace measure is defined as the set of positions that the center of the moving platform can attain with any direction of its $z$-axis making a tilt angle limited by a given value. Therefore, the orientation workspace of the Gough–Stewart platform can be kept to be symmetrical. Then, the configuration of the Gough–Stewart platform can be defined by the vector $\mathbf{x} = [x_t, y_t, z_t, \phi, \theta, \sigma]$ while $[x_t, y_t, z_t]$ represents the 3D location of the center of the moving platform and $[\phi, \theta, \sigma]$ defines the T&T angles.

The requirements that must be achieved by the Gough–Stewart platform in this case are given in Table 1. They have been fixed after discussion with some of our industrial partners. First of all, the maximum tilt workspace of the Gough–Stewart platform should cover a cube of side length $l_0 \geq 100$ mm and the range of T&T angles being $\phi \in (-\pi, \pi]$, $\theta \in [0, \pi/12]$, and $\sigma \in [0, \pi/12]$. In this workspace, several performances should be guaranteed. Thus, this cube will be called the regular dexterous workspace (RDW) of the robot [1].

Additionally, considering the reality (gain of place), the footprint of the robot must be as small as possible.

The Gough–Stewart platform optimized ought to satisfy all the following geometric and kinematic constraints throughout the RDW:

- The RDW should be free of singularity (both of the Gough–Stewart platform and the visual servoing controllers applied in this case);
- The robot positioning error should be lower than 1 mm;
- The robot orientation error should be lower than 0.01 rad;
- Some distances are constrained in order to avoid collisions or to have unpractical designs: the distance $r_b$ between the origin of the base frame $O$ and the $U$ joint position $B_i$, the distance $r_a$ between the origin of the platform frame $P_i$ and the $S$ joint $P_i$, the radius of the prismatic actuator’s $B_iP_i$ cross-section denoted as $R'$ and, finally, the camera frame location (Figure 1b). These constraints will be further detailed in Section 5.

**Table 1. Requirements of the Gough–Stewart platform.**

| Cube RDW size (side length of the cube) $l_0$ | $\geq 100$ mm |
|-----------------------------------------------|--------------|
| Tilt and Torsion angles $\phi \in (-\pi, \pi]$, $\theta \in [0, \pi/12]$, $\sigma \in [0, \pi/12]$ |
| Positioning accuracy in RDW | $\leq 1$ mm |
| Orientation accuracy in RDW | $\leq 0.01$ rad |
| No singularity in RDW of the controller of the robot |
| Constraints on geom. param. will be provided in Section 5 |

In order to get the desired 1 mm of positioning accuracy and 0.01 rad orientation accuracy specified in Table 1, we propose to apply visual servoing approaches. A single camera is chosen to be the external sensor and is mounted onto the ground in order to control the motions of the Gough–Stewart platform. The resolution of the camera is $1920 \times 1200$ pixels and the focal length is $10$ mm). The best way is to observe some image features attached to the moving platform directly with the camera. However, in some cases, it is difficult to observe the end-effector, such as the milling operations. Alternative features proposed in [20] are the cylindrical legs of the robot’s prismatic actuators. Therefore, in this...
case, three types of classical visual servoing approaches (LegBVS [20], LineBVS [28], and IMVS [19]) will be tested.

The two first controllers take the image features extracted from the observation of robot legs, while the last one will be used to observe the platform directly. The optimal design parameters of the Gough–Stewart platform for each type of controller will be found and based on the analysis of the obtained results, the best pair {robot + controller} will be determined.

In addition, several comments should be illustrated here. First, the dynamic criterion is not mentioned in these specifications. In fact, for the visual servoing, high-speed motion is not the purpose, except for a few specific scenarios [32,33]. Therefore, only the geometry and kinematics performance of the robot will be considered. Besides, a repeatability of 1 mm and orientation accuracy of 0.01 rad could also be obtained by a standard encoder-based controller. However, this paper does not aim to prove that visual servoing gets a better accuracy performance compared with standard encoder-based control. This paper aims to prove that in the condition of controlling a robot with visual servoing (or any other types of sensor-based controllers), in order to obtain the guaranteed accuracy, it is essential to optimize the robot and the controller at the same time in the design process.

In the next section, some brief recalls on visual servoing will be given in front of presenting the optimization problem formulation.

3. Recalls on Visual Servoing

In this section, a simple review on visual servoing is presented. Then, we provide some recalls on three considered approaches in particular [19,20,28].

3.1. Basics of Image-Based Visual Servoing

Image-based visual servoing is an external sensor-based controller which uses the so-called interaction matrix $L$ [5] to transform the twist $\dot{\tau}_c$ between the camera and the scene (in what follows, the superscript “c” denotes the camera frame), to the time derivative $\dot{s}$ of the vector $s$ of the visual primitives observed from the camera through the relationship:

$$\dot{s} = L(s, x) \dot{\tau}_c$$  \hspace{1cm} (1)

the components of $L$ are highly nonlinear and are a function of the image features $s$ and the robot end-effector configuration in Cartesian space $x$.

Based on (1), we can build a simple visual servoing error model:

$$\Delta s = L(s, x) \Delta x$$  \hspace{1cm} (2)

where $\Delta s$ represents a small error from the camera observation and $\Delta x$ is the corresponding positioning error of the robot end-effector in Cartesian space. As mentioned above, the matrix $L$ is a matrix whose components are nonlinear functions depending of the variables $s$ and $x$. Therefore, the matrix $L$ may meet some singularities. Positioning error models and singularities of the visual servoings [19,20,28] will be further detailed in Section 4.

Now, let us provide some recalls about the features observed in the three different types of controllers [19,20,28].

In addition, based on the kinematic relationship, one classical controller that takes the image feature $s$ as the feedback can be proposed:

$$\dot{\tau}_c = -\lambda L^+ e$$  \hspace{1cm} (3)

in which the vector $e$ stacks the error between the desired image feature $s^*$ and the current one $e = s - s^*$, $L^+$ is the pseudo-inverse of the matrix $L$, and $\lambda$ is a positive constant.

This expression can be transformed into a controller for the joint velocities:

$$\dot{q} = -\lambda J_{pins} L^+ e$$  \hspace{1cm} (4)
where $J_{\text{pinv}}$ is the inverse Jacobian matrix of the robot linking the end-effector twist to the actuator velocities, i.e., $J_{\text{pinv}} \dot{\tau}_c = \dot{q}$.

### 3.2. Recalls on Leg-Direction-Based Visual Servoing and Line-Based Visual Servoing

The legs of parallel robots are usually designed with slim and cylindrical rods, then the feature that can be extracted from the observation of the legs are their directions $\mathbf{u}_i$ (Figure 2) [20] and the line $L_i$ passing through robot link cylinder axis $i$ expressed by its Plücker coordinates $(\mathbf{u}_i, \mathbf{h}_i)$ (see definition in [11] and Figure 2).

![Figure 2. Projection of a cylinder in the image plane and the image from the camera observation.](image)

For Leg-direction-based visual servoing, we can always find the relationship between the twist of robot end-effector $\dot{\tau}_c$ and its leg direction velocity by:

$$\mathbf{u}_i = M_{ui}^T \dot{\tau}_c$$

(5)

where $M_{ui}^T$ is the interaction matrix for the leg $i$.

For Line-based visual servoing, the kinematic model aims at finding the relationship between the time variation of the Plücker coordinates $(\mathbf{u}_i, \mathbf{h}_i)$ of the robot legs and the twist of its platform [9]:

$$\begin{bmatrix} \dot{\mathbf{u}}_i^T \\ \dot{\mathbf{h}}_i^T \end{bmatrix} = M_{uhi}^T \dot{\tau}_c$$

(6)

$M_{uhi}^T$ is the interaction matrix for the leg $i$ for this type of observation.

In image plane, the contour of these cylindrical links are projected into lines $\ell^{(k)}_i$ ($k = 1, 2$) defined as the intersections of the image plane and the plane $S^{(k)}_i$ with normal $\mathbf{n}^{(k)}_i$ lying on the camera frame origin $C$ and the observed cylinder (see Figure 2). With the coordinates of the intersection points between the lines $\ell^{(k)}_i$ ($k = 1, 2$) and the image plane boundary, together with the position of the camera, we can get the vector $\mathbf{n}^{(k)}_i$ of the plan $S^{(k)}_i$. Therefore, for each cylindrical robot leg, the normal vectors $\mathbf{n}^{(k)}_i$ are information that we can extract from the camera observation. If $n$ cylinders are observed, the vector $\mathbf{s}$ of the observed features is defined as: $\mathbf{s} = [\mathbf{n}_1^{(1)T} \mathbf{n}_1^{(2)T} \ldots \mathbf{n}_n^{(1)T} \mathbf{n}_n^{(2)T}]^T$. Moreover, as we see from Figure 2, it should be mentioned that only observing part of one robot leg (the entire view of the leg is not essential) can get the intersection points of its edges and the image plan boundary.

Therefore, for leg-direction-based visual servoing, we can get the relationship between the time variation of $\mathbf{u}_i$ and the derivative of image features $[\mathbf{n}^{(1)T}_i \mathbf{n}^{(2)T}_i]^T$ with respect to time with

$$\mathbf{u}_i = J_{ui} [\mathbf{n}^{(1)}_i \mathbf{n}^{(2)}_i]^T$$

(7)
where \( J_u \) and \( J_h \) transform the time derivative of \((c^v \mathbf{n}_1^1, c^v \mathbf{n}_2^1)\) into the vector velocities of \((\dot{c^u} \mathbf{n}_1^1, \dot{c}^u \mathbf{n}_2^1)\).

To fully control the six DOF of the Gough–Stewart platform, observing a minimum of three independent legs is necessary. Therefore, when using LegBVS, we obtain the end-effector twist \( \dot{c}^v \tau_c \) with:

\[
\dot{c}^v \tau_c = J_u \mathbf{c} \mathbf{n}
\]

The matrix \( M^T_{uh} \) can be obtained by stacking the matrices of \( M^T_{ui} \) of \( k \) legs \((k = 3, \cdots, 6)\) (the way of getting the interaction matrix \( M^T_{ui} \) is presented in Appendix A). \( J_u \) is a block-diagonal matrix containing the matrix \( J_{ui} \). By using the pseudo-inverse \( M^T_{ui} \) of the matrix \( M^T_{ui} \), we have:

\[
\dot{c}^v \tau_c = M^T_{ui} J_u \mathbf{c} \mathbf{n} = L^T_{ui} \mathbf{c} \mathbf{n}
\]

In the condition of Line-based visual servoing, the end-effector twist \( \dot{c}^v \tau_c \) can be obtained from the function:

\[
\dot{c}^v \tau_c = J_u J_h \mathbf{c} \mathbf{n}
\]

The matrix \( M^T_{ui} \) is obtained by stacking the matrices \( M^T_{ui} \) of \( k \) legs \((k = 3, \cdots, 6)\) (the way of getting the interaction matrix \( M^T_{ui} \) is presented in Appendix A), \( J_u, J_h \) are block-diagonal matrices containing the matrices \( J_{ui}, J_{hi} \). Then, by using the pseudo-inverse \( M^T_{ui} \) of the matrix \( M^T_{ui} \), we have:

\[
\dot{c}^v \tau_c = M^T_{ui} \left[ J_u \right] \mathbf{c} \mathbf{n} = L^T_{ui} \mathbf{c} \mathbf{n}
\]

### 3.3. Image Moment Visual Servoing

IMVS is different from the previous ones. It is an approach based on the observation of a target \( T \) mounted on the moving platform of the robot (Figure 3). The image moments can be extracted from the image plane through the observation of the camera [19]. The target \( T \) can be a dense object defined by a set of closed contours or a discrete set of \( m \) image points [34]. For the target \( T \), we denote \( U \) the projection of the target in image plane. Then we can compute the image moment of \( U \): the moment \( m_{uv} \) of order \( w + t \) is defined by:

\[
m_{uv} = \iint_{U} u^w v^t \, dxdy
\]

where \( u \) and \( v \) are the coordinates in the image plane of any point belonging to the surface \( U \). The interaction matrix associated with any moment is provided in [19].

For a Gough–Stewart platform with six DOFs, a set of six independent moments should be selected as the image features. In this work, \( T \) is set to be a discrete model composed of three points \((A_1, A_2, A_3)\) (Figure 4). The selection of the proper image features is always a complex problem. We especially need to find six combinations of moments to control the six DOFs of the robot. The best selection of the visual servoing features is that they can be used to design a decoupled control scheme. That is try to associate each DOF to be controlled with only one visual feature. This can provide a large domain of convergence, a good behavior of the visual features, and an adequate camera trajectory. However, until now, no one found such a combination of image moments. In this paper, the objective is to obtain a sparse interaction matrix that changes slowly around the desired position and the selection of image moments is the same as it was in [34]. It has been proven that the
coordinates \(x_g, y_g\) of the center of gravity, and the area \(a = m_{00}\) of the object are the classical image features are enough to control the three translational DOFs. In addition, in order to control the rotational DOF, we need to use the object orientation \(\alpha\) and two moments \(c_1\) and \(c_2\) (see definitions in [34]). \(c_1\) and \(c_2\) have been proven to be invariant to translation and 2-D rotation. In conclusion, for image moment visual servoing applied in this case, the image feature is \(\mathbf{m} = [x_g, y_g, a, \alpha, c_1, c_2]^T\). Then we have:

\[
\mathbf{m} = \mathbf{L}_m \cdot \tau_c
\]

(14)

where \(\dot{\mathbf{m}} = [\dot{x}_g, \dot{y}_g, \dot{\alpha}, \dot{c}_1, \dot{c}_2]^T\) are the time derivatives of six image features observed. \(\mathbf{L}_m = [L_{x_g}, L_{y_g}, L_a, L_\alpha, L_{c_1}, L_{c_2}]^T\) is the interaction matrix related to the set of image moments [19,34].

Figure 3. A camera observing the robot legs.

Figure 4. Discrete model composed of three points for image moment visual servoing.

In the model for estimating the robot platform configuration based on the image features \(\mathbf{m} = [x_g, y_g, a, \alpha, c_1, c_2]^T\), the coordinates of the three points \((A_1, A_2, A_3)\) (Figure 4) are involved, as well as the camera pose. The value of these parameters will be optimized later during the design optimization process.
It should be noticed that, despite the fact that there is no explicit appearance of the robot geometric parameters in the interaction model of image moment visual servoing controller, they still have an influence on its performance: the location of the robot workspace is defined by the robot geometric parameters. If the distance between the workspace and camera location is long, the accuracy performance will be worse than if the workspace is closer to the camera location. Accordingly, we still need to optimize the robot geometric parameters in order to optimize the overall robot accuracy.

In the next section, we deal with the computation of some performance indices of the visual servoing controller.

4. Controller Performance

Concerning the requirements of positioning accuracy for the robot design, two types of controller performance will be defined and considered:

- The presence (or even proximity) controller singularities, the singularities of the interaction matrices impact both the positioning accuracy and the controller stability [4];
- The positioning error comes from the camera observation error and the interaction model of the corresponding visual servoing controller.

Then, in this section, singularities of the corresponding controllers and the positioning error models are described.

4.1. Controller Singularities

It was defined in [35] that the rank deficiency of the interaction matrix \( L \) leads to the visual servoing controller singularity. In this section, based on the study of the controllers defined in Section 3, we show the conditions of rank deficiency of the corresponding interaction matrices.

4.1.1. Leg-Based Visual Servoing Singularities

The singularity problem of the mapping between the space of the observed image features and the Cartesian space has a great influence on the accuracy of visual servoing. Thanks to the work of [36], a tool named “Hidden robot” was developed in order to simplify the study of the controller singularity problem when visual servoing is applied to the control of the parallel robot. It reduces the study of the complex singularities of the interaction matrix to the study of the singularities of the virtual parallel robot hidden in the controller. The main idea of the “Hidden robot” is to find the virtual actuators that correspond to the observation image features. For example, when we apply the Leg-direction-based visual servoing to control the Gough–Stewart platform, we choose the unit vector \( \mathbf{u}_i \) as the image feature. The unit vector in space can be parameterized by two independent coordinates (see Figure 5) that can be the angles defined by the U joint rotations. Therefore, the displacement of the U joint can be measured by the vector \( \mathbf{u}_i \). As a result, the U joint is the virtual actuator (in another way, the “hidden robot”) of the vector \( \mathbf{u}_i \). From (4), we see that the visual servoing can meet numerical issues if the matrix \( L^T \) is rank deficient and a null error vector \( \mathbf{e} \) leads to a non-null platform twist \( \mathbf{c}_c \); or the matrix \( L^T + \) is rank deficient and the controller may meet a local minimum, which means that the error \( \mathbf{e} \) is not zero but the twist \( \mathbf{c}_c \) is zero. The interaction matrix \( L \) involved in the controller gives the value of \( \mathbf{s} \) as a function of \( \mathbf{c}_c \). Therefore, \( L^T \) can be seen as the inverse Jacobian matrix of the hidden robot (moreover, \( H^+ \) is the hidden robot Jacobian matrix). Then, \( L^T \) is rank deficient only when the corresponding hidden robot comes to Type 2 singularity loci and \( L^T + \) is rank deficient only when the corresponding hidden robot comes to Type 1 singularity loci. Therefore, the hidden robot helps simplify the analysis of the interaction matrix singularity by reducing this problem to the singularity analysis of a new robot.
In [37], the problem of LegBVS controller singularities for the control of the Gough–Stewart platform has been detailed presented. The Gough–Stewart platform consists of six UPS legs. The corresponding hidden robot of the UPS leg is made of UPS legs. Since UPS legs have 2 degrees of actuation, only three legs to be observed are enough to fully control the Gough–Stewart platform when using leg direction observation [36].

The singular configurations of 3-UPS-like robots have been deeply studied in [38,39]. Type 2 singularities appear when the planes $P_1, P_2, P_3$ (whose normal directions are defined by the vectors $u_1, u_2, u_3$ and the plane $P_4$ (passing through the points $P_1, P_2, P_3$ in Figure 6) intersect in one point (which can be at infinity) (Figure 6).

Figure 5. A unit vector $u_i$ in space and its parameterization.

![Figure 5](image)

Figure 6. Example of a Type 2 singularity for a 3-UPS robot: the platform gets an uncontrollable rotation around $P_1P_2$ [37].

Singularities of LineBVS applied to the control of the Gough–Stewart platform have never been studied before. The concept of the hidden robot is to find what kind of virtual actuators correspond to the features of observation applied in visual servoing. For LineBVS, we take the Plücker coordinates of a line $L_i$ as the image feature to be observed and it can be defined from the fact that a 3D point and a 3D orientation define a unique 3D line [11]. Therefore, we should find the virtual actuators corresponding to the 3D line $L_i$. 

![Figure 6](image)
As we see from Figure 7, \( B_i \) is the 3D point and \( \mathbf{u}_i \) the unit vector, \( \mathcal{L}_i \) (\( i = 1, 2, \cdots , 6 \)) is the 3D line they define. The active \( \mathcal{U} \) joint in space is the virtual actuator that makes the vector \( \mathbf{u}_i \) move. In general, the actuated PPP chain should be added on the preceding leg links so that the point \( B_i \) can move in space. Therefore, for a \( \mathcal{UPS} \) leg, its corresponding hidden robot when using line-based visual servoing is a \( \mathcal{PPPUP} \)S leg (Figure 7). However, in the case of a Gough–Stewart platform, all the \( \mathcal{U} \) joints are fixed on the base, which means that the points \( B_i \) are fixed in space. Then the actuated PPP chain is no longer needed and the 3D lines \( \mathcal{L}_i \) passing through the robot links can be defined only by the vectors \( \mathbf{u}_i \). Therefore, the corresponding hidden robot of the Gough–Stewart platform is the same as the hidden robot when applying leg-direction-based visual servoing, the 3 − \( \mathcal{UP} \)S robot, which means that these two visual servoing controllers share the same conditions of controller singularities. Then, we suppose that in terms of controller performances, LegBVS and LineBVS are the same (which will be proven in the following Section).

![Figure 7. Corresponding hidden robot leg when the line \( \mathcal{L}_i \) in space is observed.](image)

### 4.1.2. Image Moment Visual Servoing Singularities

For IMVS, the controller singularity appears when the matrix \( \mathbf{L}_m \) is rank deficient. The expression of the matrix \( \mathbf{L}_m \) is rather complex and it is difficult to find the condition of rank deficient analytically. We should define a criterion of “proximity” to controller singularities. A list of indices that could be adapted in the analysis of robot singularity was presented in [40]. In this case, we take the inverse conditioning of the interaction matrix as the index of the controller singularity to estimate the numerical stability of the interaction matrix \( \mathbf{L}_m \).

### 4.2. Positioning Accuracy Model

#### 4.2.1. Observation Errors in the Leg-Based Visual Servoing

The positioning error models when observing the robot links in the Leg-based visual servoing approaches have been detailed and presented in [22,24]. The positioning error comes from the camera observation error of image features (For LegBVS, the features are the leg directions, for LineBVS, the features are the leg Plücker coordinates). When we use the camera to observe the robot links, the link edges are projected into the image plane into lines \( \ell_{ij}^{(k)} \) (Figure 3), which are then pixelized (Figure 8). We suppose that the error of estimation of the lines \( \ell_{ij}^{(k)} \) is due to a random shift of ±0.5 px in the pixels corresponding to
the intersections of the $\ell^{(k)}_i$ with the image plane boundary (points $P^{(1)}_{ik}$ and $P^{(2)}_{ik}$ in Figure 8) in this case.

As we presented in Section 3, the image features that we use in the leg-based visual servoing are the vectors $n^{(k)}_i$ (characterizing the line $\ell^{(k)}_i$). Thus, we can find the mapping relating the time derivatives of the vectors $n^{(k)}_i$ to the derivative of $P^{(1)}_{ik}$, $P^{(2)}_{ik}$ with respect to time:

$$
\dot{n}^{(k)}_i = J_{n_{ik}} \begin{bmatrix} \dot{P}^{(1)}_{ik} \\ \dot{P}^{(2)}_{ik} \end{bmatrix}
$$

(15)

Then we get the error model,

$$
\Delta n^{(k)}_i = J_{n_{ij}} \begin{bmatrix} \Delta P^{(1)}_{ik} \\ \Delta P^{(2)}_{ik} \end{bmatrix}
$$

(16)

where $\Delta n^{(k)}_i$, $\Delta P^{(1)}_{ik}$ and $\Delta P^{(2)}_{ik}$ are the small variations of the vectors $n^{(k)}_i$, $P^{(1)}_{ijk}$ and $P^{(2)}_{ijk}$ respectively. Based on the controllers presented in Section 3, for the image features $s = [\dot{n}^{(1)}_1^{T}, \dot{n}^{(2)}_1^{T}, \ldots, \dot{n}^{(1)}_n^{T}, \dot{n}^{(2)}_n^{T}]^{T}$, we have

$$
\Delta s = J_{s} \Delta p
$$

(17)

where $\Delta s$ stands for the small variations of the image features $s$, $\Delta p$ contains all errors $\Delta P^{(1)}_{ik}$ and $\Delta P^{(2)}_{ik}$.

In this case, the camera observation noise is set to be $\pm 0.5$ pixel, which is a typical noise for cameras. Thus every component of vector $\Delta p^{(1)}_{ik}$ and $\Delta p^{(2)}_{ik}$ can take the values $+0.5$ or $-0.5$. With the help of Equation (17), we can get the observation error model for LegBVS and LineBVS written under the generic form:

$$
\Delta x = L_{p} \Delta p
$$

(18)

where $L_{p} = L^{+} J_{n}$.

![Figure 8. Error for the observation of a line.](image)

4.2.2. Observation Errors in the Image Moment Visual Servoing

Image moments are calculated from the coordinates of the points belonging to the projection on the image plane of the object observed. We set $(x_1, y_1), (x_2, y_2), (x_3, y_3)$ as
to be the coordinates of the projection of the three points \( A_1, A_2, A_3 \) (Figure 9) in pixel. Then we have

\[
\frac{\partial m}{\partial t} = \frac{\partial m}{\partial Q} \frac{\partial Q}{\partial t} = S \frac{\partial Q}{\partial t}
\]

(19)

where \( Q = [x_1 x_2 x_3 y_1 y_2 y_3]^T \) and \( S \) is the matrix which transforms the time derivatives of the set of image moments \( m \) to the time derivatives of the coordinates of the points projected to the pixel plane.

Thus, Equation (14) can be written as the form

\[
\tau = L_m^+ m = L_m^+ S Q
\]

(20)

We estimate that the error of estimation of each component of \( Q \) to be \( \pm 0.5 \) pix (see Figure 9) for the location of each point projected in the image plane. Then the error model of the image moment visual servoing controller can be written in the form:

\[
\Delta x = L_m^+ S \Delta Q
\]

(21)

4.2.3. Positioning Accuracy

For the Gough–Stewart platform, we have \( \Delta x = [\Delta t_x \Delta t_y \Delta t_z \Delta w_x \Delta w_y \Delta w_z] \) with \([\Delta t_x \Delta t_y \Delta t_z]\) being the translation errors along the three axes and \([\Delta w_x \Delta w_y \Delta w_z]\) being the rotation errors around the three axes. Then the positioning error and the orientation error are defined as in [30]:

\[
E_t = \sqrt{\Delta t_x^2 + \Delta t_y^2 + \Delta t_z^2}
\]

(22)

and the orientation error is defined as

\[
E_w = \sqrt{\Delta w_x^2 + \Delta w_y^2 + \Delta w_z^2}
\]

(23)

In the next section, the optimal design problem for the Gough–Stewart platform will be formulated.

5. Optimal Design Procedure

In this section, the design procedure developed in order to obtain the optimal parameters of the Gough–Stewart platform together with the parameters of the controllers are described.

5.1. Design Variables

Robot design parameters: As we presented in Section 2, the Gough–Stewart platform can be defined by the following geometric parameters: \( r_a, r_b, \alpha_0, \alpha_1, \alpha_2 \) (Figure 1c) \( (r_a = ||OP_i||, r_b = ||OB_i||, \alpha_0 = \angle x'P_iP_0, \alpha_1 = \frac{1}{2}\angle B_1B_0B_2, \alpha_2 = \frac{1}{2}\angle P_1P_2P_3) \). All these
parameters have an effect on the size of robot workspace and the physics performance, as well as on the controller performance. In addition, when LegBVS and LineBVS are applied, the radius of the cylindrical distal links of the Gough–Stewart platform also influence the positioning accuracy [22], thus the radius of the cylindrical distal links \( R_i \) \((i = 1, 2, \cdots, 6)\), denoted as \( R \) (see Figure 3), is a decision variable of the optimization process. When image moment visual servoing is applied, the coordinates of the discrete three points model \([x_1 y_1 x_2 y_2 x_3 y_3]\) (in moving platform frame \(X'Y'Z'\)) defining the configuration of the model (Figure 4) affect the controller interaction model. They must be optimized when dealing with image moment visual servoing.

Controller design parameters: The configuration of the camera is normally parameterized by six independent parameters and it affects the controller interaction model. In order to observe the robot (both the robot legs and the end-effector) in a symmetrical way:

- The camera frame orientation is set to be parallel to the robot fixed frame;
- The camera origin is imposed to stay on a vertical line passing through \( O((x_c, y_c) \) of the camera frame origin set at \((0, 0))\).

Additionally, some other variables that we used in the optimal design process need to be defined: \( L \) is the length of the prismatic actuator \( B_i P_i \) \((L = \|B_i P_i\|, i = 1, 2, \cdots, 6)\). \( l_0 \) is the dimensions of the side length of the cube RDW (see Table 1) [1].

Design variables: Based on the explanations above, two different sets of design variables (grouped in a vector \( y \)), depending of the types of controllers are defined:

- For the leg-based controllers,
  \[
  y = [r_a, r_b, a_0, a_1, a_2, z_c, R]^T; 
  \]
- For the moment-based controllers,
  \[
  y = [r_a, r_b, a_0, a_1, a_2, z_c, x_1, x_2, x_3, y_1, y_2, y_3]^T. 
  \]

5.2. Objective Function

As mentioned in Section 2, the robot should be as compact as possible. The footprint of the Gough–Stewart platform is evaluated by the radius \( r_b \) of its base. Therefore, the optimization problem is formulated in order to minimize the value of \( r_b \).

5.3. Constraints

The constraints provided in Section 2 are reviewed here. Throughout the RDW, the following geometric and kinematic constraints must be satisfied:

- The RDW should be free of singularity (both of the robot and the controller): Singularities of the controllers are detailed and presented in Section 4.1. In this case, we used the inverse condition number of the interaction matrix \( L \), denoted as \( \kappa^{-1}(L) \). In the RDW, we want to have
  \[
  \kappa^{-1}(L) > 10^{-3} \tag{24} 
  \]
  The “mechanics” singularity of the Gough–Stewart platform is different. This problem is complex and was studied decades ago [1,41–44]. In [45,46], a kinetostatic approach taking account of the force transmission was proposed to determine the singularity-free zones of a parallel robot. When the pressure angle is close to 90 degrees, the parallel robot is close to a singular configuration. Therefore, we calculated the pressure angles \( \beta = [\beta_1, \cdots, \beta_6]^T \) for all the six robot legs of the Gough–Stewart platform [45,46]. In the RDW, we want to have
  \[
  \beta_i > 80^\circ \quad i = 1, \cdots, 6 \tag{25} 
  \]
- The value of the robot positioning accuracy ought to be lower than 1 mm and the orientation accuracy should be lower than 0.02 rad. The positioning error model is defined in Section 4.2. The error models are linear in terms of the observation error, the maximal positioning error \( E_{\max} = \max \| E_i \| \) and the maximal orientation error \( E_{w, \max} = \max \| E_w \| \) of the robot will be found at one the corners of the hyper-
polyhedron defining the observation errors [47]. The repeatability constraint can be formulated as:

\[
E_{t\text{max}} \leq 1 \text{ mm,} \\
E_{w\text{max}} \leq 0.02 \text{ rad (26)}
\]

- The discrete three points \(A_1, A_2, A_3\) should be within the moving platform of the Gough–Stewart platform.
- The end-effector should be within the view of the camera: ensuring that all the robot distal legs can be observed when using leg-based visual servoing, as well as the three points \(A_1, A_2, A_3\) can be observed when using image moment visual servoing.
- Several distances or angles are constrained in order to avoid collisions or to have unpractical designs. The values of these constraints are given here:

\[
0.4 \text{ m} \leq L \leq 0.76 \text{ m,} \\
0.1 \text{ m} \leq r_a \leq 0.3 \text{ m} \leq r_b \leq 0.5 \text{ m,} \\
r_a < 0.9 \times r_b, \\
-\frac{\pi}{6} \leq a_0 \leq \frac{\pi}{6}, \\
0 \leq a_1 \leq \frac{\pi}{9}, \\
0 \leq a_2 \leq \frac{\pi}{4}, \\
-0.2 \text{ m} \leq z_c \leq 0.3 \text{ m,} \\
0.01 \text{ m} \leq R \leq 0.03 \text{ m,}
\]

The aforementioned RDW throughout which all the constraints (24)–(27) must be satisfied should cover a cube of side length \(l_0 \geq 100 \text{ mm}\) and the range of T&T angles being \(\phi \in (-\pi, \pi]\), \(\theta \in [0, \pi/12]\), and \(\sigma \in [0, \pi/12]\). The algorithm of calculating the size of the Largest Regular Dexterous Workspace (LRDW) is detailed and presented in [27] and is adapted in this case for getting the cubic LRDW among the RDW of the manipulator for a given decision variable vector \(y\).

We denote \(l_{LRDW}\) the side length of the cubid LRDW whose T&T angles range are \(\phi \in (-\pi, \pi]\), \(\theta \in [0, \pi/12]\), and \(\sigma \in [0, \pi/12]\). We make sure that all constraints (24)–(27) are obligatory true throughout the LRDW, then only one is needed to replace all the other ones, which is defined by:

\[
l_{LRDW} \geq 0.1 \text{ m (28)}
\]

### 5.4. Problem Formulation and Optimization Results

For designing a compact Gough–Stewart platform with the detailed specifications given in Table 1, the following optimization problem is formulated:

\[
\begin{align*}
\text{minimize} \quad & r_b \\
\text{over} \quad & y \\
\text{subject to} \quad & l_{LRDW} \geq 100 \text{ mm}
\end{align*}
\]

where the definition of \(y\) is given in Section 5.1.

As introduced in Section 3, observing three legs is enough to fully control the Gough–Stewart platform when leg-based visual servoing controllers are applied. In this case, as a matter of comparison, we will optimize the geometric parameters of the Gough–Stewart platform when observing only three legs ([Case 1]: observing robot links \(B_1P_1, B_3P_3, B_5P_5\)) and observing all the six legs ([Case 2]: observing robot links \(B_1P_1, B_2P_2, B_3P_3, B_4P_4, B_5P_5, B_6P_6\)) for leg-based visual servoing.
The optimization algorithm presented above is then applied to the design of the Gough–Stewart platform, for each of the three controllers defined in Section 3. These optimization problems have been solved by means of the ‘active-set’ algorithm implemented in the MATLAB \textit{fmincon} function. A \textit{multistart} algorithm, combined with random initial points initialized by a Genetic Algorithm, was also used in order to increase the chances to reach the global minima. The optimal design results are given in Table 2 and illustrated in Figures 10–12.

Figure 10. Gough–Stewart platform optimized using LineBVS [Case 1].

Figure 11. Gough–Stewart platform optimized using LineBVS [Case 2].
Figure 12. Gough–Stewart platform optimized using image moment visual servoing.

Table 2. Design parameters and value of the objective function for the chosen controller.

|                  | LegBVS [Case 1] | LineBVS [Case 1] | LegBVS [Case 2] | LineBVS [Case 2] | IMVS |
|------------------|-----------------|------------------|-----------------|------------------|------|
| $r_a [\text{m}]$| 0.2054          | 0.2054           | 0.1402          | 0.1402           | 0.1600 |
| $r_b [\text{m}]$| 0.3000          | 0.3000           | 0.3000          | 0.3000           | 0.3000 |
| $a_0 [\text{rad}]$| −0.4256        | −0.4256          | −0.4243         | −0.4243          | 0.2668 |
| $a_1 [\text{rad}]$| 0.2318          | 0.2318           | 0.2298          | 0.2298           | 0.1986 |
| $a_2 [\text{rad}]$| 0.1424          | 0.1424           | 0.6927          | 0.6927           | 0.2406 |
| $z_c [\text{m}]$| −0.0523         | −0.0523          | −0.0551         | −0.0551          | 0.1204 |
| $R [\text{m}]$   | 0.0197           | 0.0197           | 0.0199          | 0.0199           | N/A  |
| $x_1 [\text{m}]$ | N/A             | N/A              | N/A             | N/A              | −0.1311 |
| $x_2 [\text{m}]$ | N/A             | N/A              | N/A             | N/A              | 0.1303 |
| $x_3 [\text{m}]$ | N/A             | N/A              | N/A             | N/A              | −0.1044 |
| $y_1 [\text{m}]$ | N/A             | N/A              | N/A             | N/A              | −0.0870 |
| $y_2 [\text{m}]$ | N/A             | N/A              | N/A             | N/A              | −0.0839 |
| $y_3 [\text{m}]$ | N/A             | N/A              | N/A             | N/A              | 0.0976 |
| $r_b [\text{m}]$| 0.3000          | 0.3000           | 0.3000          | 0.3000           | 0.3000 |

As we see from the results of optimization, in terms of the footprint of the robot, the Gough–Stewart platform designed based on the LegBVS, LineBVS, and IMVS are close from each other and the differences are almost negligible. Especially, for robots designed for leg-based visual servoing controllers, the geometric parameters of the robot are the same under the same observing condition (Case 1 and Case 2). This result proves our hypothesis proposed in Section 4.1.1. The coordinates of points $B_i$ are constant since the points $B_i$ are fixed, then the time derivative of $h_i$ and $u_i$ are linearly dependent, which means that the LegBVS and LineBVS share the same controller performance.
In the next section, we will perform co-simulations with ADAMS and Simulink to test the robot accuracy performance.

6. Results Cross-Validations through Simulations
6.1. Simulation Method

In order to validate the optimization results and test the robot accuracy performance, the co-simulations are performed within a connected ADAMS-Simulink environment (Figure 13). Five Gough–Stewart platform models with the optimal geometric parameters obtained from the optimal design process (one model per controller) are created in the software ADAMS.

Real-time data (block “Data acquisition”) of the ADAMS simulator are extracted:
- For LegBVS and LineBVS, we extract the coordinates of the points $P_i$ and $B_i$ (Figure 1b);
- For IMVS, the coordinates of the three points $A_1$, $A_2$ and $A_3$ (Figure 4) are extracted.

The scheme of the co-simulation is illustrated in Figure 13. The frequency of the simulation is set to be 200 Hz. Real-time data of mechanical models are the output of ADAMS and are sent to Simulink. In Simulink, the model of the camera is created and the real-time data are projected to pixel plane by the camera model to rebuild image features. The $\pm 0.5$ pixel random noise related to the observation errors presented in Section 4 is added in the pixel plane. Then the image features with noise become the feedback of the control loop to generate the velocity command. The velocity command is the input of ADAMS and is used to control the motion of the robot mechanical model.

![Figure 13. Co-simulation control scheme of Gough–Stewart platform.](image)

The RDW of the Gough–Stewart platform is a cube whose side length is 100 mm, and the orientation workspace is set based on the T&T angles $\phi \in (-\pi, \pi]$, $\theta \in [0, \pi/12]$, $\sigma \in [0, \pi/12]$. A home position $T_1$ and nine desired positions (including $T_1$) within the LRDW are defined in Table 3 with respect to the center of the LRDW. For each position, three orientation pose are defined with respect to $[\phi \theta \sigma]^T$: Pose 1 $[0, 0, 0]^T$, Pose 2 $[\pi/2, \pi/12, \pi/12]^T$, and Pose 3 $[-\pi/2, \pi/12, \pi/12]^T$. Therefore, for each robot, a total of 27 desired poses are selected in the co-simulations.

Then, each robot is driven from their home pose to the desired poses with the dedicated controller. All their positioning accuracies and orientation accuracies are recorded during the co-simulation.

| Point | Coordinate [m] | Point | Coordinate [m] | Point | Coordinate [m] |
|-------|----------------|-------|----------------|-------|----------------|
| $T_1$ | (0,0,0)        | $T_4$ | (-0.05,-0.05,0.05) | $T_7$ | (0.05,0.05,-0.05) |
| $T_2$ | (0.05,0.05,0.05) | $T_5$ | (-0.05,0.05,0.05) | $T_8$ | (0.05,-0.05,-0.05) |
| $T_3$ | (0.05,-0.05,0.05) | $T_6$ | (-0.05,0.05,-0.05) | $T_9$ | (-0.05,-0.05,-0.05) |
Additionally, in order to test the robustness of the accuracy of model with geometry errors, the same co-simulations were operated with the error added in model. The models we added errors on joints to are defined as below: we add a random error on the location of the joint $B_i$ on the base of the robot, the distance between the accurate joint $B_i$, and the joint with error $B_i'$, denoted as $l_{B_iB_i'}$ ($l_{B_iB_i'} = 0.1 \times r_i$) (see the red parts of Figures 10–12).

In the next step, The designed robot prototypes were controlled with another controller, different from the one dedicated during the design process, for verifying the original purpose of performing control-based design. In what follows, for brevity, only the result of LineBVS applied to the robot designed for the image-based moments will be given here.

Results are shown and analyzed in the next subsection.

### 6.2. Simulation Results

In this section, we denote as:

- **[Case A]**: the Gough–Stewart platform optimized for LineBVS ([Model 1]) in ([Case 1]) and the error-added robot mechanical model ([Model 2]) controlled with their dedicated controller, robot links $B_iP_i$ ($i = 1, 3, 5$) are observed;
- **[Case B]**: the Gough–Stewart platform optimized for LineBVS ([Model 3]) in ([Case 2]) and the error-added robot mechanical model ([Model 4]) controlled with their dedicated controller, all six robot links $B_iP_i$ ($i = 1, 2, 3, 4, 5, 6$) are observed;
- **[Case C]**: the Gough–Stewart platform optimized for IMVS ([Model 5]) and the error-added robot mechanical model ([Model 6]) controlled with its dedicated controller;
- **[Case D]**: the Gough–Stewart platform optimized for IMVS ([Model 5]) controlled with the LineBVS, all six links $B_iP_i$ ($i = 1, 2, 3, 4, 5, 6$) are observed.
- **[Case E]**: the Gough–Stewart platform optimized for LineBVS ([Model 3]) controlled with the IMVS.

Since we have proved that the LegBVS and LineBVS have the same control performance for the Gough–Stewart platform and the geometric parameters of the robot designed for these two controllers are the same under the same observation condition, we only perform the co-simulations for the robot that controlled with LineBVS. We played each simulation for five seconds and recorded the positioning error. The simulation results show that the robot converges at around 0.5 s, then the moving platform oscillates around the desired pose due to the simulated observation noise. For all the simulation motions from home position to the desired poses in Case A to Case E, the maximum positioning and orientation error along the time are recorded: for point $T_{kj}$ ($k$ for the position $k = 1, \ldots, 9$, $j$ for the pose $j = 1, 2, 3$) simulated in case $\alpha$ ($\alpha = A, B, C, D$), this maximum positioning error is denoted as $\delta p_{kj\alpha}$ and the maximal orientation error is denoted as $\delta \theta_{kj\alpha}$. Then, all the results are summarized in Table 4: for each case and different model, max, min, and mean value of positioning error $\delta p_{kj\alpha}$ and orientation error $\delta \theta_{kj\alpha}$ obtained for $k = 1, \ldots, 9$ and $j = 1, 2, 3$ are shown for a given value of $\alpha$.

Studying the results, we see that the robot Model 5 in Case C leads to minimal positioning error and orientation error. For robots in Case A and Case B, the mean value is very close to the requested value of 1 mm. However, there are some points in the workspace for which the error is slightly upper this limit (maximal error of 1.24 mm in both cases). In fact, the positioning accuracy model applied during the optimal design process (Section 5) in order to estimate the controller performance was really simple. It was thus the source of inaccuracies of positioning error estimation during the optimal design process. However, even with this simplistic model, the maximal robot positioning error (1.24 mm) is only slightly upper the threshold of 1 mm while their mean values stay close to 1 mm.

Additionally, the measured orientation error obtained from all the cases are far lower than the requested 0.01 rad. The results obtained from the models with geometry errors are similar to the results obtained from the accurate models, which proves the robustness of the accuracy of models when applying the visual servoing controllers.
Table 4. Results of co-simulation in terms of end-effector accuracy: min, max, standard deviation, and mean values for the error recorded on the tested 24 points.

| Case | Max Positioning Error [mm] | Min Positioning Error [mm] | Mean Positioning Error [mm] | Max Orientation Error [rad] | Min Orientation Error [rad] | Mean Positioning Error [rad] |
|------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|
| A ([Model 1]) | 1.24 | 0.94 | 1.03 | 4.5 × 10⁻⁴ | 2.2 × 10⁻⁴ | 3.1 × 10⁻⁴ |
| A ([Model 2]) | 1.23 | 0.96 | 1.05 | 4.5 × 10⁻⁴ | 2.0 × 10⁻⁴ | 3.5 × 10⁻⁴ |
| B ([Model 3]) | 1.12 | 0.91 | 0.99 | 4.0 × 10⁻⁴ | 1.9 × 10⁻⁴ | 2.8 × 10⁻⁴ |
| B ([Model 4]) | 1.24 | 0.99 | 1.01 | 4.0 × 10⁻⁴ | 2.1 × 10⁻⁴ | 3.0 × 10⁻⁴ |
| C ([Model 5]) | 0.63 | 0.28 | 0.38 | 4.3 × 10⁻⁴ | 2.2 × 10⁻⁴ | 2.8 × 10⁻⁴ |
| C ([Model 6]) | 0.66 | 0.29 | 0.42 | 4.6 × 10⁻⁴ | 2.6 × 10⁻⁴ | 3.0 × 10⁻⁴ |
| D ([Model 5]) | 1.56 | 1.37 | 1.44 | 5.0 × 10⁻⁴ | 3.3 × 10⁻⁴ | 4.4 × 10⁻⁴ |
| E ([Model 5]) | 1.47 | 1.21 | 1.39 | 4.3 × 10⁻⁴ | 2.3 × 10⁻⁴ | 3.4 × 10⁻⁴ |

We then study the results of [Case D] and [Case E], which are the most important. For the Gough–Stewart platform optimized for IMVS but controlled with the LineBVS, the mean error is far bigger than the requested value of 1 mm, and the maximal error even grows up to 1.56 mm. For the Gough–Stewart platform optimized for LineBVS but controlled with the IMVS, the mean error is 1.39 mm, and the maximal error grows to 1.47 mm. These positioning errors are bigger than the requested value of 1 mm and they are worse than the results of [Case B] and [Case C]. These results confirm that it is necessary to optimize a robot for a dedicated controller. In other words, the control-based design of [robot+controller] helps ensure the vision-based control accuracy performance.

Another problem, which is the most interesting, is that the discrete three points model we obtained from the optimal design when using IMVS to form a triangle (Triangle 1) which is not a regular triangle. Therefore, in order to study why it is such a configuration, we create discrete three points whose configuration is a regular triangle (Triangle 2). The coordinates of the three points (with respect to the moving platform frame \(x'O'y'\)) are \(A_{1r}, A_{2r}, A_{3r}\) are \((0, 0.222)\) m, \((0.192, -0.111)\) m, and \((-0.192, 0.111)\) m. The same noise was added on the projection of the points in pixel to see the variation of the set of image moments \(m\). For Triangle 1 and Triangle 2, in terms of the image moments \([x_g, y_g, a]\), the variations are almost the same. However, in terms of the image moments \([a, c_1, c_2]\), the differences are huge: for Triangle 1, the variations of \([a, c_1, c_2]\) are \([0.01, 0.08, 0.08]\), while the variations of the image moments \([a, c_1, c_2]\) for Triangle 2 are \([1.6, 20, 400]\) (the results of \(a\) are illustrated in Figures 14 and 15). In addition, we performed the same co-simulation as we did for Model 5 in Case C, but the target is changed to the new three discrete points model (Triangle 2) in IMVS. The simulation results show that the maximal positioning error comes to 1.6 mm and the maximal orientation error comes to \(6.0 \times 10^{-4}\) rad, while the corresponding results are 0.63 mm and \(4.3 \times 10^{-4}\) rad when observing the three discrete points model Triangle 1. The results prove that the configuration of the discrete points model has an influence on the observation of image moments and affects the controller accuracy performance. As a result, it is necessary to perform topology optimization on the configuration of the shape of the target observed during the design process.
7. Conclusions

In the work presented above, a novel advanced optimal design methodology “control-based design” is performed in order to design a Gough–Stewart robot with the best accuracy performance of the pair {robot + controller}. We have proven that the controller performance (accuracy, singularity) is affected by the robot geometric design parameters. Thus, in the design process of a robot, it is necessary to find the optimized geometric parameters of the robot that will allow the best performance of the pair {robot + controller}.

Three different classical types of visual servoing controllers: LegBVS, LineBVS, and IMVS were proposed to be applied on the Gough–Stewart platform. Positioning error models considering the camera observation error were developed based on the study of these three types of controllers. In addition, in order to avoid the instability issues, the singularities of these controllers were analyzed for purpose of avoiding the controller singularities. In the next step, the design optimization problem for getting the optimal geometric parameters and the placement of the camera for the Gough–Stewart platform has been formulated for each type of controller. Then, co-simulations between ADAMS and Simulink for the Gough–Stewart platforms optimized for the three controllers were performed. The results showed that the robots designed for these three visual servoing controllers had a similar size (robots designed for LegBVS and LineBVS share the same size). The robot designed for IMVS had a better positioning accuracy compared with the other two robots optimized for LegBVS and LineBVS. Especially, the co-simulation results show that when one controller is applied on a robot designed for another one, the positioning error performance was no longer guaranteed, confirming the importance of the control-based design approach. In the future, experimental works on real prototypes are necessary in order to verify the simulation results.
Author Contributions: Conceptualization, M.Z. and C.H.; data curation, M.Z. and D.G.; methodology, M.Z. and S.S.; validation, M.Z. and S.S.; formal analysis, M.Z. and D.G.; writing—original draft preparation, M.Z. and C.H.; writing—review and editing, D.G.; supervision, M.Z.; project administration, C.H.; funding acquisition, D.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (61803285, 62001332); the National Defense Pre-Research Foundation of China (H04W201018).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: We acknowledge the contribution of Jun Qi and Chuan Lu for their help in software, investigation, resources, and visualization.

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

Appendix A. Leg-Based Visual Servoing Interaction Matrix

Appendix A.1. Leg-Direction-Based Visual Servoing Interaction Matrix

Leg-direction-based visual servoing kinematic aims at finding the relationship between the time variation of the unit vector $\mathbf{u}_i$ of the robot legs and the twist of its platform.

The leg direction $\mathbf{u}_i$ extracted from the observation of the robot leg $\mathbf{B}_i\mathbf{P}_i$ (Figure 1b) is selected as the feature to do the visual servoing. We have

$$\mathbf{u}_i = (\mathbf{P}_i^G - \mathbf{B}_i^G) / L_i$$

where $L_i$ is the length of the prismatic actuator $\mathbf{B}_i\mathbf{P}_i$. $\mathbf{P}_i^G$ and $\mathbf{B}_i^G$ represent the coordinates with respect to the global frame. $\mathbf{P}_i^G (i = 1 \cdots 6)$ can be obtained from:

$$\mathbf{P}_i^G = \mathbf{T} + \mathbf{R}(\phi, \theta, \sigma)\mathbf{P}_i^L$$

$\mathbf{T}$ and $\mathbf{R}$ denote the position of the center of the mobile frame and its rotation matrix in the global frame. $\mathbf{P}_i^L$ denotes the coordinate of the point $\mathbf{P}_i$ in local frame.

From Equation (A2), we have the vision-based kinematics of the Gough–Stewart platform expressed in the global frame:

$$L_i \mathbf{u}_i = \mathbf{T} + \mathbf{R}(\phi, \theta, \sigma)\mathbf{P}_i^L - \mathbf{B}_i^G$$

$$\dot{\mathbf{u}}_i = \frac{1}{L_i} \frac{1}{dt} \mathbf{P}_i^L \dot{\mathbf{B}}_i + \frac{L_i}{L_i} \mathbf{u}_i$$

By inserting the interaction matrix associated to a 3D point [16], we get

$$\mathbf{u}_i = -\frac{1}{L_i} [\mathbf{I}_3 - \mathbf{[B]}_x] \tau - \frac{L_i}{L_i} \mathbf{u}_i$$

where $[\cdots]_x$ is the antisymmetric matrix associated to the cross product [16]. With the help of the inverse Jacobian matrix, we can obtain the relationship between each $\mathbf{u}_i$ and $\tau$.

$$\mathbf{u}_i = \mathbf{M}_{ui}^T \tau$$

$$\mathbf{M}_{ui}^T = -\frac{1}{L_i} ([\mathbf{I}_3 - \mathbf{u}_i\mathbf{u}_i^T] [\mathbf{I}_3 - \mathbf{[B]}_x])$$

Then, the interaction matrix $\mathbf{M}_{ui}^T$ can be obtained by stacking the matrices $\mathbf{M}_{ui}^T$ of k legs ($k = 3, 4, 5, 6$).
Appendix A.2. Line-Based Visual Servoing Interaction Matrix

Line-based visual servoing kinematics aims at finding the relationship between the time variation of the Plücker coordinates \((\mathbf{u}_i, \mathbf{h}_i)\) of the robot legs and the twist of its platform.

From the definition of the Plücker coordinates, we have \(\mathbf{h}_i = \mathbf{D} \times \mathbf{u}_i\) where \(\mathbf{D}\) is the position of any point on the line passing through the center of the cylindrical leg. Then we have

\[
\mathbf{h}_i = \mathbf{D} \times \mathbf{u}_i + \mathbf{D} \times \mathbf{u}_j
\]

(A8)

For the Gough-Stewart platform, the U-joints \(\mathbf{B}_i\) \((i = 1, 2 \cdots 6)\) (Figure 1b) are all fixed on the base platform, then the Equation (A8) can be written in the form

\[
\mathbf{h}_i = \mathbf{B}_i \times \mathbf{u}_i + \mathbf{B}_i \times \mathbf{u}_j = \mathbf{B}_i \times \mathbf{u}_i
\]

(A9)

With the help of Equation (A6), the Equation (A9) can be written in the matrix form

\[
\mathbf{h}_i = [\mathbf{B}_i]_x \times \mathbf{M}_{u}^T \tau = \mathbf{M}_{u}^{\theta} \tau
\]

(A10)

\(\mathbf{M}_i\) is the interaction matrix related to the vector \(\mathbf{u}_i\). Therefore, for a line \(\mathcal{L}_i\), we have

\[
\begin{bmatrix}
\mathbf{u}_i \\
\mathbf{h}_i
\end{bmatrix} =
\begin{bmatrix}
\mathbf{M}_{u}^{\theta} \\
\mathbf{M}_{u}^{\theta}
\end{bmatrix}
\tau = \mathbf{M}_{u}^{\theta} \tau
\]

(A11)

Then, the interaction matrix \(\mathbf{M}_{u}^{\theta}\) can be obtained by stacking the matrices of \(\mathbf{M}_{u}^{\theta}\) of \(k\) legs \((k = 3, 4, 5, 6)\).

References
1. Merlet, J.P. Parallel Robots; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2006; Volume 128,
2. Merlet, J.P. 2012. Available online: http://www-sop.inria.fr/members/Jean-Pierre.Merlet/merlet.html (accessed on 9 June 2021).
3. Huang, H.; Yang, C.; Chen, C.P. Optimal robot-environment interaction under broad fuzzy neural adaptive control. IEEE Trans. Cybern. 2020, 51, 3824–3835. [CrossRef] [PubMed]
4. Chaumette, F.; Hutchinson, S.; Corke, P. Visual servoing. In Springer Handbook of Robotics; Springer: Berlin/Heidelberg, Germany, 2016; pp. 841–866.
5. Chaumette, F.; Hutchinson, S. Virtual Visual Servoing: A framework for real-time augmented reality. IEEE Trans. Autom. Mag. 2006, 13, 82–90. [CrossRef]
6. Yang, C.; Wu, H.; Li, Z.; He, W.; Wang, N.; Su, C.Y. Mind Control of A Robotic Arm with Visual Fusion Technology. IEEE/ASME Trans. Mechatron. 2021, 26, 3265–3275. [CrossRef]
7. Li, P.; Shu, T.; Xie, W.F.; Tian, W. Dynamic Visual Servoing of A 6-RSS Parallel Robot Based on Optical CMM. J. Intell. Robot. Syst. 2021, 102, 40. [CrossRef]
8. Dallej, T.; Andreff, N.; Mezouar, Y.; Martinet, P. 3D pose visual servoing relieves parallel robot control from joint sensing. In Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, Beijing, China, 9–15 October 2006; pp. 4291–4296.
9. Andreff, N.; Dallej, T.; Martinet, P. Image-based visual servoing of a gough—Stewart parallel manipulator using leg observations. Int. J. Robot. Res. 2007, 26, 677–687. [CrossRef]
10. Hamel, T.; Mahony, R. Visual servoing of an under-actuated dynamic rigid-body system: An image-based approach. IEEE Trans. Robot. Autom. 2002, 18, 187–198. [CrossRef]
11. Andreff, N.; Espiau, B.; Horaux, R. Visual servoing from lines. Int. J. Robot. Res. 2002, 21, 679–699. [CrossRef]
12. Wu, B.; Zhong, J.; Yang, C. A Visual-Based Gesture Prediction Framework Applied in Social Robots. IEEE/CAA J. Autom. Sin. 2021, 9, 510–519. [CrossRef]
13. Lu, Z.; Wang, N.; Yang, C. A Constrained DMPs Framework for Robot Skills Learning and Generalization from Human Demonstrations. IEEE/ASME Trans. Mechatron. 2021, 26, 3265–3275. [CrossRef]
14. Peng, G.; Chen, C.; Yang, C. Neural Networks Enhanced Optimal Admittance Control of Robot-Environment Interaction Using Reinforcement Learning. IEEE Trans. Neural Netw. Learn. Syst. 2021. [CrossRef]
15. Merckel, L.; Nishida, T. Multi-interfaces approach to situated knowledge management for complex instruments: First step toward industrial deployment. AI Soc. 2010, 25, 211–223. [CrossRef]
16. Martinet, P.; Gallice, J.; Khadraoui, D. Vision based control law using 3d visual features. In Proceedings of the World Automation Congress, WAC’96, Robotics and Manufacturing Systems, Montpellier, France, 28–30 May 1996; pp. 497–502.
17. Marchand, É.; Chaumette, F. Virtual Visual Servoing: A framework for real-time augmented reality. Comput. Graph. Forum 2002, 21, 289–297. [CrossRef]
18. Yang, C.; Peng, G.; Li, Y.; Cui, R.; Li, Z. Neural Networks Enhanced Adaptive Admittance Control of Optimized Robot-Environment Interaction. *IEEE Trans. Cybern.* **2018**, *49*, 2568–2579. [CrossRef]

19. Chaumette, F. Image moments: A general and useful set of features for visual servoing. *IEEE Trans. Robot.* **2004**, *20*, 713–723. [CrossRef]

20. Andreff, N.; Marchadier, A.; Martinet, P. Vision-based control of a Gough-Stewart parallel mechanism using legs observation. In Proceedings of the 2005 IEEE International Conference on Robotics and Automation, Barcelona, Spain, 18–22 April 2005; pp. 2535–2540. [CrossRef]

21. Kaci, L.; Boudaud, C.; Martinet, P. Control-based Design of a Five-bar Mechanism. In *New Trends in Mechanism and Machine Science*; Springer: Berlin/Heidelberg, Germany, 2017; pp. 303–311.

22. Kaci, L.; Boudaud, C.; Briot, S.; Martinet, P. Elastostatic Modelling of a Wooden Parallel Robot. In *Computational Kinematics*; Springer: Berlin/Heidelberg, Germany, 2018; pp. 53–61.

23. Briot, S.; Martinet, P.; Rosenzveig, V. The hidden robot: An efficient concept contributing to the analysis of the controllability of parallel robots in advanced visual servoing techniques. *IEEE Trans. Robot.* **2015**, *31*, 1337–1352. [CrossRef]

24. Zhu, M.; Chrétte, A.; Briot, S. Control-based Design of a DELTA robot. In Proceedings of the Symposium on Robot Design, Dynamics and Control, Sapporo, Japan, 20–24 September 2020; pp. 204–212.

25. Michel, H.; Rives, P. Singularities in the Determination of the Situation of a Robot Effector from the Perspective View of 3 Points. Ph.D. Thesis, INRIA Sophia Antipolis, Biot, France, 1993.

26. Pascual-Escudero, B.; Nayak, A.; Briot, S.; Kermognet, O.; Martinet, P.; El Din, M.S.; Chaumette, F. Complete Singularity Analysis for the Perspective-Four-Point Problem. *Int. J. Comput. Vis.* **2021**, *129*, 1217–1237. [CrossRef]

27. Germain, C.; Caro, S.; Briot, S.; Wenger, P. Optimal Design of the IRSBot-2 Based on an Optimized Test Trajectory. In Proceedings of the ASME International Design Engineering Technical Conferences & Computers & Information in Engineering Conference, Portland, OR, USA, 4–7 August 2013.

28. Vignolo, A.; Briot, S.; Philippe, M.; Chen, C. Comparative analysis of two types of leg-observation-based visual servoing approaches for the control of the five-bar mechanism. In Proceedings of the 2014 Australasian Conference on Robotics and Automation (ACRA 2014), Melbourne, Australia, 2–4 December 2014.

29. Stewart, D. A platform with six degrees of freedom. *Proc. Inst. Mech. Eng.* **1965**, *180*, 371–386. [CrossRef]

30. Bonev, I.A.; Ryu, J. Orientation workspace analysis of 6-DOF parallel manipulators. In Proceedings of the International Design Engineering Technical Conferences and Computers and Information in Engineering Conference. American Society of Mechanical Engineers, Las Vegas, NV, USA, 12–16 September 1999; Volume 19715, pp. 281–288.

31. Blaise, J.; Bonev, I.; Monsarrat, B.; Briot, S.; Lambert, J.M.; Perron, C. Kinematic characterisation of hexapods for industry. *Int. J. Robot. Int.* **2010**, *37*, 79–88. [CrossRef]

32. Fusco, F.; Kermognet, O.; Martinet, P. Integrating Features Acceleration in Visual Predictive Control. *IEEE Robot. Autom. Lett.* **2020**, *5*, 5197–5204. [CrossRef]

33. Dahmouche, R.; Andreff, N.; Mezouar, Y.; Ait-Aider, O.; Martinet, P. Dynamic visual servoing from sequential regions of interest acquisition. *Int. J. Robot. Res.* **2012**, *31*, 520–537. [CrossRef]

34. Tahri, O.; Chaumette, F. Point-based and region-based image moments for visual servoing of planar objects. *IEEE Trans. Robot.* **2005**, *21*, 1116–1127. [CrossRef]

35. Hutchison, S.; Hager, G.D.; Corke, P.I. A tutorial on visual servo control. *IEEE Trans. Robot. Autom.* **1996**, *12*, 651–670. [CrossRef]

36. Briot, S.; Martinet, P. Minimal representation for the control of Gough-Stewart platforms via leg observation considering a hidden robot model. In Proceedings of the 2013 IEEE International Conference on Robotics and Automation (ICRA), Karlsruhe, Germany, 6–10 May 2013; pp. 4653–4658.

37. Briot, S.; Rosenzveig, V.; Martinet, P.; Özgür, E.; Bouton, N. Minimal representation for the control of parallel robots via leg observation considering a hidden robot model. *Mech. Mach. Theory* **2016**, *106*, 115–147. [CrossRef]

38. Ben-Horin, P.; Shoham, M. Singularity analysis of a class of parallel robots based on Grassmann–Cayley algebra. *Mech. Mach. Theory* **2006**, *41*, 958–970. [CrossRef]

39. Caro, S.; Moroz, G.; Gayral, T.; Chablat, D.; Chen, C. Singularity analysis of a six-dof parallel manipulator using grassmann-cayley algebra and groebner bases. In *Brain, Body and Machine*; Springer: Berlin/Heidelberg, Germany, 2010; pp. 341–352.

40. Merlet, J.P. Jacobian, manipulability, condition number, and accuracy of parallel robots. *J. Mech. Des.* **2006**, *128*, 199–206. [CrossRef]

41. Fichter, E.F. A Stewart platform-based manipulator: General theory and practical construction. *Int. J. Robot. Res.* **1986**, *5*, 157–182. [CrossRef]

42. Ben-Horin, P.; Shoham, M. Application of Grassmann–Cayley algebra to geometrical interpretation of parallel robot singularities. *Int. J. Robot. Res.* **2009**, *28*, 127–141. [CrossRef]

43. St-Onge, B.M.; Gosselin, C.M. Singularity analysis and representation of the general Gough-Stewart platform. *Int. J. Robot. Res.* **2000**, *19*, 271–288. [CrossRef]

44. Hunt, K.H.; Hunt, K.H.; Hunt, K.H. *Kinematic Geometry of Mechanisms*; Oxford University Press: New York, NY, USA, 1978; Volume 7.

45. Arakelian, V.; Briot, S.; Glazunov, V. Improvement of functional performance of spatial parallel manipulators using mechanisms of variable structure. In Proceedings of the 12th IFToMM World Congress, Besançon, France, 17–21 June 2007.
46. Arakelian, V.; Briot, S.; Glazunov, V. Increase of singularity-free zones in the workspace of parallel manipulators using mechanisms of variable structure. *Mech. Mach. Theory* **2008**, *43*, 1129–1140. [CrossRef]

47. Briot, S.; Pashkevich, A.; Chablat, D. Optimal Technology-Oriented Design of Parallel Robots for High-Speed Machining Applications. In Proceedings of the 2010 IEEE International Conference on Robotics and Automation (ICRA), Anchorage, AK, USA, 3–7 May 2010.