Available online at www.sciencedirect.com

ScienceDirect

Procedia Manufacturing 55 (2021) 64–71

Procedia MANUFACTURING

www.elsevier.com/locate/procedia

30th International Conference on Flexible Automation and Intelligent Manufacturing (FAIM2021)
7-10 September 2021, Athens, Greece.

A simulation driven development framework for parallel kinematics

Lukas Bath[a,*], Thorsten Schüppstuhl[a]

[a]*Institute of Aircraft Production Technology, Hamburg University of Technology, Denickestraße 17, Hamburg 21073, Germany*

Abstract

This work presents a novel software tool to design, analyze and simulate parallel robots. It considers any desired constraints, requirements and load scenarios, while being able to be used by non-experts. This is comprehensively shown for one kind of the *Gough*/*Stewart*-platform kinematics class. First, the inverse kinematics is derived, which allows for the interactive exploration of the workspace, as well as other properties, prior to building the machine. In addition, the forward kinematics are implemented iteratively. The validation of the simulation is done by presenting a low cost, six degree of freedom robot for laboratory applications, being designed using this framework. Here, both the inverse and forward kinematics are used in real time to allow for precise and correct movement, detailed in-process measurements, such as process loads, along with overload protection. The approach fosters the efficient selection of existing as well as the design of custom robots of this kind.

Keywords: Parallel kinematics; Simulation; Interactive development; Laboratory hexapod; Forward transformation

1. Introduction

Numerous applications for parallel manipulators exist such as simulators, machine tools or arbitrary laboratory scenarios [1]. Hereto, these robots offer special benefits which include high accuracy, stiffness and speed while being capable of handling high loads [2]. Serial kinematics, on the contrary, often fail to keep up in these aspects, while being easier to design and providing larger workspaces [3].

However, the options being commercially available are limited. Finding an optimal parallel robot within reasonable time and price range for a certain laboratory use case is not easy. Furthermore, these machines are complex and much research exists on various challenges these kinematics introduce. The focus of this work lies on six degree of freedom parallel mechanisms, also referred to as *Gough*/*Stewart*-platforms or hexapod robots. Finding suitable properties of a machine according to requirements such as accuracy and repeatability, workspace, speed and load capability is difficult without deeper understanding. So far, only few tools exist to aid in the design process. Neither do they include all the relevant aspects nor are they meant to be used by non-experts.

In this work, a novel simulation driven development framework is introduced, that can be used to effortlessly tune all relevant parameters of hexapod robots, while taking any desired constraints, requirements and load scenarios into consideration. This interactive design process is illustrated for one special class of hexapod robots, which is especially useful for applications where elongated workspaces are needed. The interactive simulation framework supports starting from the design towards the application of these robots and fosters both the interactive evaluation of existing machines as well as the development of customized versions with no compromises. Along with fully featured workspace analyses, feasibility studies conducted this way may thus also include arbitrary trajectory evaluations, for instance. Obviously, this logic can also be used for the control of such robots.

It follows a review of related work, a characterization of the studied robot and an introduction to the inverse kinematics.

2. Background

In contrast to serial mechanisms, parallel kinematics have closed kinematic chains. Any pose represented using the former can easily be determined by following the chained transformations of the joints, which is referred to as forward kinematics. This is not possible for the latter.

* Corresponding author. Tel.: +49-40-42878-2022; fax: +49-40-42731-4551.

E-mail address: lukas.bath@tuhh.de (Lukas Bath).

2351-9789
10.1016/j.promfg.2021.10.010

Here, the inverse kinematics is of simpler form, which is particularly useful for controlling the target pose of such a robot. The error of establishing a desired pose is relatively low compared to serial robots, where the errors of the individual joints accumulate. Parallel robots have high strength-to-weight ratios, since the load is near uniformly distributed over the parallel actuators.

The workspaces of these robots can be planar or spatial, with up to three or six degrees of freedom, respectively. The workspace can be assessed using geometric, discrete or numerical approaches [1].

The kinematic chains, referred to as legs, consist of multiple joints and at least one actuator. In the case of one actuator per chain, six legs are required and in the case of two actuators per chain, three legs are sufficient to allow for six degrees of freedom of the platform [2]. Joints can be of type revolute (R), prismatic (P), universal (U) and ball-and-socket (S) and actuators of type revolute (R) or prismatic (U) [1], for example.

The first platform with six degrees of freedom was invented by *Gough* in 1947 [4] and rediscovered by *Stewart* in 1965 [5].

Typical applications comprise simulators, machine tools, positioning devices, handling technique, testing machines [1] or dentistry/medicine [6].

In this work, the focus lies on one distinct kind of parallel robot which is referred to as *Gough*/*Stewart*-platform. These robots in their original form consist of a planar tool platform and a planar base, both being connected by six legs that are adjustable in length and attached through joints and therefore providing six degrees of freedom [2]. Both hexagons must not be equal as singularities are likely to occur in that case. Numerous other configurations exist.

There are three types of singularities, which are studied in various publications [7, 8].

Various attempts regarding the common problem of the forward kinematics have been made by analyzing solution spaces [9], by providing simplified (analytical) forms [10], by developing numeric real-time capable approximation methods [11, 12] or by the introduction of virtual legs [13].

In many publications, numerical simulations were conducted, nearly none of them presenting a dedicated software tool for this, except for [14] for cable robots and [15] for many parallel manipulators which comes closest, but was designed for 195 different configurations with legs adjustable in length and based on *Matlab*. The 195 different configuration arise from freely combining the therein defined four leg types. All other approaches perform manual simulations using respective mathematics software [16, 17, 18, 7].

In [15], various software tools are mentioned along with the developed one. These either do not cover the kind of mechanism studied in this work or study fewer degrees of freedom.

Alongside software tools within the scope of scientific research, few commercial but free to use tools exist that are provided by manufacturers of respective robots[1]. Both commercial tools offer only limited workspace calculation and visualization capabilities. The former constrains either the position or the orientation during the calculation and the latter supports only the calculation of positions within the workspace. Both are limited to one kind of kinematic model and to the commercially available machines of that class. They support basic load investigations, no trajectory simulations and the overall investigation options are rather limited.

Alongside dedicated research and commercial tools for the analysis of parallel robots, another option is the use of generic CAD or multi-body simulation programs [19, 20]. However, especially the latter are rather meant to be used for dynamic analyses than for the complete computation and analysis of the entire workspace, as it is focused in this work. In addition, the underlying algorithms are primarily optimized for solving large nonlinear equation systems of arbitrary geometric problems instead of processing huge quantities of different poses. All studied scientific, commercial or generic tools require significant expert knowledge and may even require to model the architecture of the desired kinematic from scratch.

Regarding the same kind of parallel robot that is studied in this work, previous research addressed performance indices [21], forward dynamics [22], and the general control and application [23].

Merlet and *Daney* give an overview of design strategies for parallel mechanisms [24]. These include the selection of a particular architecture, the modelling of the kinematic and the reduction of the number of freely adjustable design parameters. Eventually, an optimal, or in most cases, only an appropriate set of parameters is determined using one out of several different approaches.

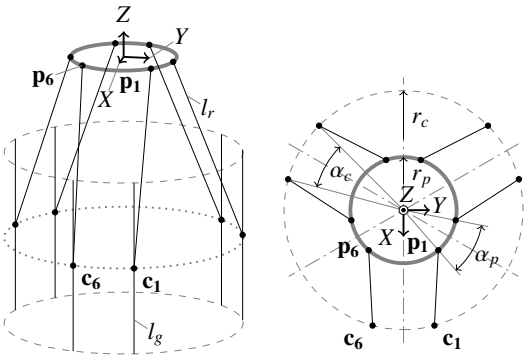
3. Characterization

Similar to most other variations amongst *Gough*/*Stewart*-platforms, the kinematic studied in this work consists of a planar and rigid tool platform (illustrated in bold gray in Fig. 1). Underneath, it holds six universal joints $\mathbf{p_i}$ with zero backlash that are regularly placed in pairs, with their center points being on a radius r_p while separated by angle α_p. Each of them is connected through rods of fixed length l_r to another universal joint $\mathbf{c_i}$ at each rod's bottom end. The center points of these joints are placed on a radius r_c, separated by angle α_c and connected via double row ball bearings inside carriages with six ball screw linear modules of length l_g being arranged in parallel. The joints are attached such that they are unbent and collinear with the rods when the platform is at its rest position.

In contrast to most other kinematics of this class, this variant's rods are of fixed length and slide on linear guides instead of being fixed to a rigid base. At an early stage, the conventional *Gough*/*Stewart*-platform's workspace was found to be too limited in height and orientation for the specimen available. For this reason, the development of the simulation framework was only continued with respect to the above described kinematic. The conventional kind will be added in the future again.

The robots' legs' kinematic chains in this work can be expressed as 6-PRUU or in a simplified form as 6-PUS.

[1] PI Hexapod Simulation Tool (www.physikinstrumente.com), HexaViz (www.newport.com)



(a) Isometric view (b) Top view

Fig. 1. Kinematic model

4. Inverse kinematics

In the following, the inverse kinematics are described to obtain the required carriage joints' positions $\mathbf{c_i}$ for a given pose $\mathbf{a}$ of the platform. This is used for the control of the robot.

The kinematic is defined in *Cartesian* space. While at its rest position, the platform's center matches the global origin. The platform's pose can be described as $\mathbf{a} = [\mathbf{t}, \mathbf{r}]^T$ constituting of the translation $\mathbf{t} = [x, y, z]^T$ as well as the *Euler* rotation angles $\mathbf{r} = [\alpha, \beta, \gamma]^T$. Using the respective rotation matrix $\mathbf{R}$ these are applied according to the intrinsic ZXZ convention.

The platform's orientation shall be composed of a tilt towards any desired XY-direction along with a rotation around its own normal. According to the ZXZ convention, it is hereto first rotated around Z by α. X' then represents the axis around which the platform is tilted by β and is orthogonal to the desired XY-tilting direction. Last, the platform is rotated around Z''. Since it has already been rotated around Z in the first step, this needs to be compensated by using $\gamma = \gamma' - \alpha$ as the rotation angle. Like this, it is effectively rotated around its normal by γ', only.

Typically, there do apply practical limitations to these tilt and rotation angles. The kinematic presented in this work is designed to allow for $0° \leq \beta \leq 20°$ and $|\gamma'| \leq 20°$.

Nomenclature

Geometric constants

r_c	radius of carriage joints
α_c	separation angle between pairs of carriage joints
r_p	radius of platform joints
α_p	separation angle between pairs of platform joints
l_r	length of rods
l_g	effective length of guide rails

Constraints

γ_j	joints' angular limit
f	platform load of forces
t	platform load of torques
f_c	allowed carriage load
f_f	allowed frame load

Miscellaneous

α	1st *Euler* platform angle around Z
β	2nd *Euler* platform angle around X'
γ	3rd *Euler* platform angle around Z''
$\mathbf{c_i}$	center points of carriages' universal joints
$\mathbf{p_i}$	center points of platform's universal joints
$\mathbf{t}$	platform's translation
$\mathbf{T}$	tool center point transformation
$\mathbf{R}$	*Euler* rotation matrix
$\mathbf{E}$	coefficients of linear equation system
$\mathbf{f_r}$	rods' forces
$\mathbf{r_d}$	rods' vectors
$\mathbf{l_i}$	levers of platform torque
$\mathbf{r_i}$	cross product of rods and levers
$\mathbf{l}$	platform's design load
$\mathbf{f_a}$	actuators' forces
$\mathbf{f_f}$	frame's forces
$\mathbf{A}$	partial derivatives of the platform's pose
$\mathbf{l_t}$	true, in-process load of the platform

Six rods connect pairs of platform joints $\mathbf{p_i}$ and carriage joints $\mathbf{c_i}$ with $i = 1...6$ while

$$\mathbf{p_i} = \mathbf{T} \cdot \begin{cases} \begin{pmatrix} r_p \cdot \cos(\frac{\pi}{3} \cdot i - \frac{\alpha_p}{2}) \\ r_p \cdot \sin(\frac{\pi}{3} \cdot i - \frac{\alpha_p}{2}) \\ 0 \\ 1 \end{pmatrix} & \text{, for } i = 1, 3, 5 \\ \begin{pmatrix} r_p \cdot \cos(\frac{\pi}{3} \cdot (i-1) + \frac{\alpha_p}{2}) \\ r_p \cdot \sin(\frac{\pi}{3} \cdot (i-1) + \frac{\alpha_p}{2}) \\ 0 \\ 1 \end{pmatrix} & \text{, for } i = 2, 4, 6 \end{cases} \tag{1}$$

with

$$\mathbf{T} = \begin{pmatrix} \mathbf{R} & \mathbf{t} \\ 0 & 1 \end{pmatrix} \tag{2}$$

and

$$\mathbf{c_i} = \begin{pmatrix} \mathbf{c_{i_x}} \\ \mathbf{c_{i_y}} \\ -\sqrt{l_r^2 - (\mathbf{p_{i_x}} - \mathbf{c_{i_x}})^2 - (\mathbf{p_{i_y}} - \mathbf{c_{i_y}})^2} + \mathbf{p_{i_z}} \\ 1 \end{pmatrix} \tag{3}$$

with

$$\begin{pmatrix} \mathbf{c_{i_x}} \\ \mathbf{c_{i_y}} \end{pmatrix} = \begin{cases} r_c \cdot \begin{pmatrix} \cos(\frac{\pi}{3} \cdot (i-1) + \frac{\alpha_c}{2}) \\ \sin(\frac{\pi}{3} \cdot (i-1) + \frac{\alpha_c}{2}) \end{pmatrix} & \text{, for } i = 1, 3, 5 \\ r_c \cdot \begin{pmatrix} \cos(\frac{\pi}{3} \cdot i - \frac{\alpha_c}{2}) \\ \sin(\frac{\pi}{3} \cdot i - \frac{\alpha_c}{2}) \end{pmatrix} & \text{, for } i = 2, 4, 6 \end{cases}. \tag{4}$$

In order to establish any desired platform pose, (3) is the essential equation to position the carriages accordingly.

5. Simulation

The simulation developed in this work can be used to evaluate existing parallel robots in detail or to design and dimension custom robots. It is based on the previously derived inverse kinematics and supplemented by the checking of several constraints as well as load scenarios. According to the desired simulation resolution in terms of discrete translation and rotation steps the workspace is assessed iteratively. In sequential order, it

- gathers an initial pose space to be evaluated further,
- excludes geometrically invalid poses and
- evaluates reachable poses under defined process load.

5.1. Initial pose space

The initial pose space itself is obtained in multiple steps to be further analyzed from geometry and load related points of views. In sequential order,

- the desired translative and rotative resolutions are set,
- a set of equally distributed positions within a certain cylinder is created,
- the angular boundaries to test for α, β, γ are set and
- the previously created set of positions is extended by rotations.

The translative resolution defines the voxel size the cylindrical space is divided into. All vertices' positions are tested. The cylinder's radius should be $\leq \mathbf{a_{c_i}}$ and can be further reduced to prevent collisions of the platform with the robots frame, for instance, when moving towards the borders of the working space. The height of the cylinder is determined by l_g and its center Z-position is given by the plane of $\mathbf{a_{i_z}}$ while the platform is at its rest position. Example angular ranges are: $0 \leq \alpha \leq 360°, 0° \leq \beta \leq 90°, |\gamma| \leq 45°$. Each position within the previously created set is extended by a fully combined set of rotations defined by the three angles according to the convention mentioned in the previous section. If it is already known that the parallel mechanism is not required or able to exceed certain angles, the ranges can be further limited, for instance: $0 \leq \alpha \leq 360°, 0° \leq \beta \leq 20°, |\gamma| \leq 20°$.

A *Cartesian* coordinate system is chosen here since an alternative cylindrical coordinate system would introduce a decreasing accuracy with increasing radius under the assumption that positions in the space are expressed with a fixed number of decimal places for angles. On the other hand, it could have been taken advantage of the threefoldly symmetry of the kinematics working space using a cylindrical coordinate system and therefore, could have saved one third of computation efforts during the simulation.

Further simplifications can be made by multiplying the computation results for the inner layers of the simulated working space from that point on when no carriage touches the lower end of the linear modules anymore and until the point in the other z-direction when at least one carriage touches the upper boundary of the linear module again. This inner space is all the same due to the parallel arrangement of the linear modules. Hereto, the theoretical minimum of maximum positions of the carriages need to be checked against the linear modules' lower and upper boundaries. So the inner layers can be skipped during calculation.

5.2. Geometrically valid poses

Each of the previously gathered poses is tested against several conditions with the objective to reduce the amount of poses being required to be tested with further, more computationally expensive tests. In sequential order,

- checking the reachability of each pose under consideration of leg lengths, carriage positions and joint boundary angles,
- rejecting poses in case of singularities and
- rejecting poses in case of collisions.

For checking the reachability according to leg lengths the condition in (5) must be met.

$$\sqrt{(\mathbf{p_{i_x}} - \mathbf{c_{i_x}})^2 - (\mathbf{p_{i_y}} - \mathbf{c_{i_y}})^2} < \mathbf{l_g} \tag{5}$$

All carriages must stay within the guide rails' ranges:

$$\mathbf{c_{0_z}} - \frac{l_g}{2} \leq \mathbf{c_{i_z}} \leq \mathbf{c_{0_z}} + \frac{l_g}{2} \tag{6}$$

It must be made sure, that all joints' maximum allowed angles are not exceeded as in (7).

$$\gamma_j \geq \begin{cases} \arccos \dfrac{\mathbf{r_{d_i}} \cdot \mathbf{r_{d_{0_i}}}}{|\mathbf{r_{d_i}}||\mathbf{r_{d_{0_i}}}|}, \text{ for } \mathbf{c_i} \\ \arccos \dfrac{\mathbf{r_{d_i}} \cdot (\mathbf{R} \cdot \mathbf{r_{d_{0_i}}})}{|\mathbf{r_{d_i}}||(\mathbf{R} \cdot \mathbf{r_{d_{0_i}}})|}, \text{ for } \mathbf{p_i} \end{cases} \tag{7}$$

With $r_{d_{0_i}}$ being the rod directions in the robot's rest position and r_{d_i} being the current rod directions.

Singularities are avoided by limiting the initial working space to a cylinder, which prevents mirror configurations. Furthermore, Type I and II singularities, as described in [7], are prevented by making sure that the rods are never perpendicular with respect to the platform's plane, nor the linear guide rails and never coincide with the platform's plane, nor being parallel with the same. There is another type III kind of singularity which is equivalent to type II here, since type I can be avoided by the control [8].

Using the inverse kinematics for the control of the robot brings the advantage that practically only one axes configuration for each pose exists. This is guaranteed by applying constraints, by neglecting existing mirror configurations outside of the defined workspace and by precautionary avoiding singularities.

In order to check for collisions, the platforms three dimensional model as well as models of tools or actuators mounted on the platform are considered and checked for collisions with the robots frame.

A percentage of the reachability at each position is computed and is defined as the number of reachable orientations divided by the total number of tested orientations. It can also be in the

form of two separate percentage values, the first one giving a percentage with respect to absolute minimum angular ranges that are to be fulfilled at every position and the second one ranging beyond that minimum until a second threshold which can be considered as the theoretical maximum. Suitable poses can be filtered and color encoded in the visualization (see Fig. 2).

5.3. Valid poses under defined load

For each pose $\mathbf{a}$ the respective platform joints' positions $\mathbf{p_i(a)}$ as well as the carriage joints' $\mathbf{c_i(a)}$ positions are computed. All six normalized directions of the rods are then obtained in (8) such that the force and torque load's contribution to the rod forces can be determined. This is done using *Gauss* elimination. To get the torque load's contribution, the distances from the center $\mathbf{t}$ of the platform towards the platform's joints are considered as additional levers $\mathbf{l_i}$ of the torque in (9). The cross product $\mathbf{r_l}$ of these with the rod directions is computed in (10) and used for the equilibrium of moments. According to the individual platform orientation the load vectors of force $\mathbf{l_f}$ and torque $\mathbf{l_t}$ are rotated accordingly and composed into one vector in (11). $\mathbf{r_d}$ and $\mathbf{r_l}$ as well as the transformed load vectors are then composed into a linear system of equations, representing the system's equilibrium of forces and moments (12), in order to obtain the rod forces $\mathbf{f_r}$.

$$\mathbf{r_d} = \frac{\mathbf{c_i(a)} - \mathbf{p_i(a)}}{\|\mathbf{c_i(a)} - \mathbf{p_i(a)}\|} \tag{8}$$

$$\mathbf{l_i} = \mathbf{p_i(a)} - \mathbf{t} \tag{9}$$

$$\mathbf{r_l} = \mathbf{l_i} \times \mathbf{r_d} \tag{10}$$

$$\mathbf{l} = \begin{pmatrix} \mathbf{R(a)} \cdot \mathbf{l_f} \\ \mathbf{R(a)} \cdot \mathbf{l_t} \end{pmatrix} \tag{11}$$

$$\mathbf{l} = \mathbf{E} \cdot \mathbf{f_r} \tag{12}$$

with

$$\mathbf{E} = \begin{pmatrix} \mathbf{r_{d_{1_x}}} & \dots & \mathbf{r_{d_{6_x}}} \\ \mathbf{r_{d_{1_y}}} & \dots & \mathbf{r_{d_{6_y}}} \\ \mathbf{r_{d_{1_z}}} & \dots & \mathbf{r_{d_{6_z}}} \\ \mathbf{r_{l_{1_x}}} & \dots & \mathbf{r_{l_{6_x}}} \\ \mathbf{r_{l_{1_y}}} & \dots & \mathbf{r_{l_{6_y}}} \\ \mathbf{r_{l_{1_z}}} & \dots & \mathbf{r_{l_{6_z}}} \end{pmatrix} \tag{13}$$

The respective linear actuator forces as well as resulting frame forces can be obtained from $\mathbf{f_r}$ as $\mathbf{f_a} = \mathbf{f_{r_z}}$ and $\mathbf{f_f} = \sqrt{f_{r_{ix}}^2 + f_{r_{iy}}^2}$. The simulation also outputs the maximum actuator as well as frame forces.

5.4. Implementation

The simulation is implemented in the form of a web application and is based on different frameworks such as *Vue/Vuetify* for the user interface, *Three.js* for the 3D visualization and *math.js*. It is capable of executing computations in multiple threads by the use of web workers.

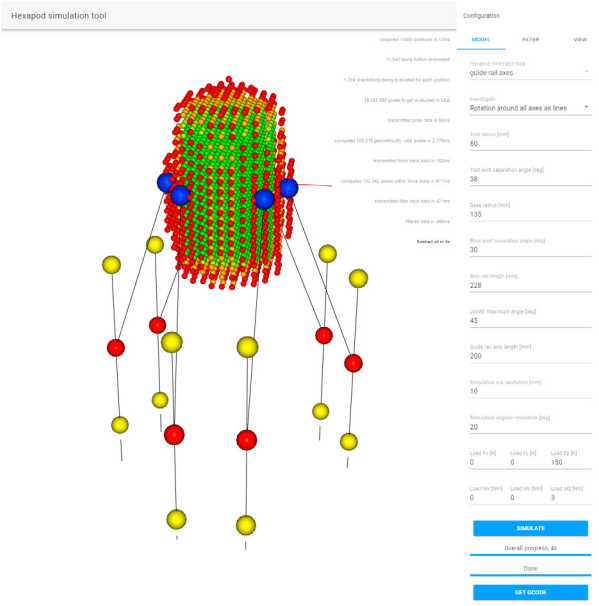


Fig. 2. The user interface of the simulation tool

The user interface as shown in figure 2 allows for tuning all the relevant parameters like geometrical constants, constraints, simulation resolutions, loads as well as angular and load related thresholds and trajectory and collision checks. The view can interactively be altered and interacted with.

5.5. Workspace analysis

Various tools can be used for workspace and layer inspection as well as trajectory simulation, as can be seen in figures 2 and 4.

Any desired trajectories may be simulated as shown in figure 3. The trajectory is visualized in the form of sequential single poses represented by colored spheres. Green spheres represent fully valid poses. Yellow poses represent exceeded load limits and red poses are geometrically impossible or not within the available workspace, respectively. Each pose represented by a sphere includes a coordinate system (X = red, Y = green, Z = blue), indicating the platform's orientation.

Seven different workspace definitions can be distinguished [1], while this work mainly focuses on the *total orientation workspace*. This definition includes all positions that can be reached with all orientations among a set of ranges of platform angles α, β and γ. Here, in addition to geometrical constraints also load conditions are considered.

5.6. Parallel kinematic design workflow

Using the interactive parallel mechanism simulation tool, certain steps (figure 5) may be followed in order to obtain the kinematic model including dimensions for building the robot:

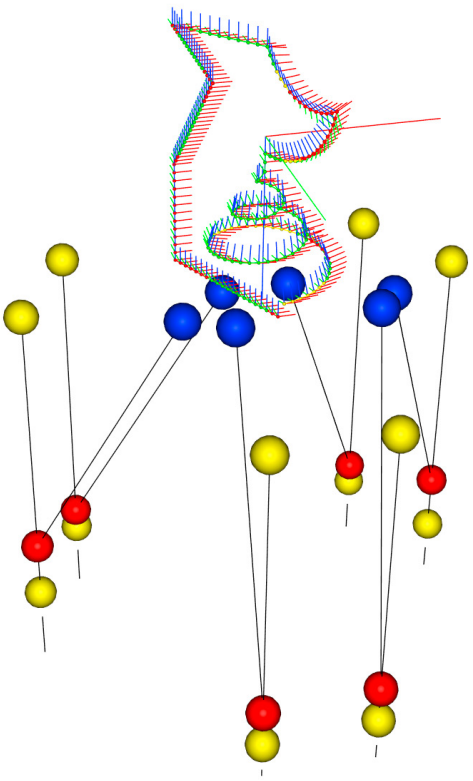
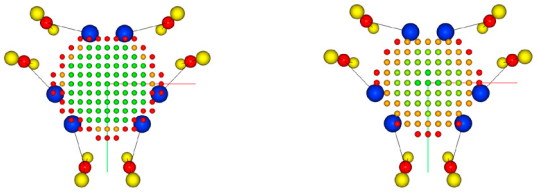


Fig. 3. Simulating arbitrary trajectories



(a) Layer at Z=10mm (b) Layer at Z=70mm

Fig. 4. The designed low-cost robot

1. gather requirements such as desired working volume, platform rotation capabilities, process loads as well as movement speeds and needed accuracies
2. define r_c and l_g to be larger than the desired workspace, make r_p about half of r_c, α_c and α_r near 30° and $l_r \approx 2 \cdot r_c$ and verify with acceptable overall build volume
3. input objected process load, allowed actuator forces as well as desired simulation resolution
4. provide further constraints (joints, collision models)
5. simulate the workspace
6. filter and explore the workspace using the various supporting aids

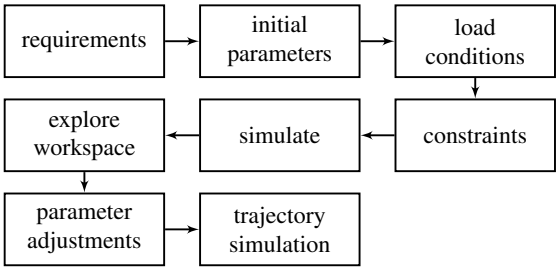


Fig. 5. Parallel robot design workflow

7. make adjustments to the kinematics parameters
8. make use of features like trajectory simulation

6. Forward Kinematics

For a given set of carriage positions and the respective joints' positions c_{0_i} the unknown pose $\mathbf{a}$ of the platform is to be determined. There exists no analytical solution to this problem. Instead, an iterative approach can be used for the localization of the platform and for the mapping of rod forces to platform load, for instance.

In order to achieve that, a linear system of equations needs to be solved for $\mathbf{a}$:

$$c_{0_i} = \mathbf{A} \cdot c_i(\mathbf{a}) \tag{14}$$

with

$$\mathbf{A} = \begin{pmatrix} \frac{\partial c_{1_z}}{\partial x} & \cdots & \frac{\partial c_{1_z}}{\partial \gamma} \\ \vdots & \ddots & \vdots \\ \frac{\partial c_{6_z}}{\partial x} & \cdots & \frac{\partial c_{6_z}}{\partial \gamma} \end{pmatrix} \tag{15}$$

Hereto, an initial guess for $\mathbf{a}$ is employed, for example $\mathbf{a} = [1, 1, 1, 0, 0, 0]^T$. This can be almost any pose but it must be ensured that $x, y, z \neq 0$ to allow the algorithm to converge appropriately. Like this, it can be assured that each pose stays within the same out of the various working spaces this kinematic has in theory.

With each iteration of the algorithm the computed error is cumulated to the previously introduced starting value. The algorithm can be aborted if the residues of all components of the pose $\mathbf{a}$ and all carriage positions c_i fall below certain thresholds or if a certain number of iterations is exceeded.

In the presented work the algorithm converges after at most six iterations with a residue of less than $0.1mm$.

As the robot moves, for each new pose to be computed iteratively, the previously computed pose may be used as a starting value to reduce the amount of required iterations.

For the implementation in this work, *Newton-Raphson* as well as *Gauss* elimination method are used, referring to a similar approach in [12].

7. Mapping of process forces

In this work, six one dimensional force transducers are included as part of the rods. However, their measurements cannot be used directly. In order to obtain the linear actuator forces or the distinct platform load, it is required to perform transformations based on the platform's pose **a** with the use of the forward kinematics, derived in the previous section.

The actuators' forces can be obtained in a simple way from the measured rods' forces as

$$f_{a_i} = f_{r_{z_i}}. \tag{16}$$

The platform's load is computed inversely compared to the simulation of forces. The platform's pose **a** is needed here. In order to find the load **l** for the known (measured) rod forces $\mathbf{f_r}$, matrix **E** is inverted and holds the equilibrium equations for each rod. These six linear equations are solved using *Gauss* method. Priorly, it held the equilibrium equations of forces and moments.

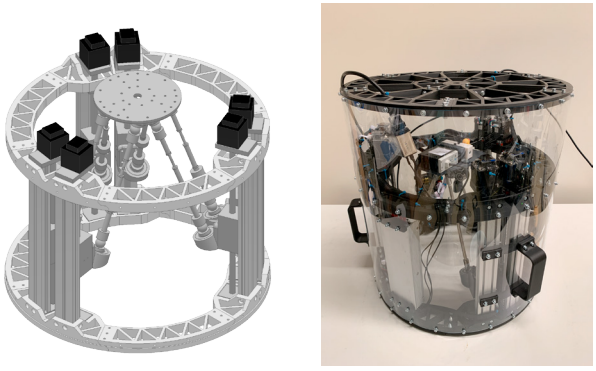
$$\mathbf{f_r} = \left(f_{r_1} \dots f_{r_6}\right)^T = \mathbf{E}^{-1} \cdot \mathbf{l} \tag{17}$$

Since the platform may be rotated and **l** will include the already rotated load, this rotation needs to be reverted in order to get the true load $\mathbf{l_t}$, with respect to the already tilted platform.

$$\mathbf{l_t} = \mathbf{R}(\mathbf{a})^T \cdot \mathbf{l} \tag{18}$$

8. Realization

The following briefly illustrates the low-cost implementation.



(a) CAD-model (b) Hexapod prototype

Fig. 6. The realized low-cost robot

8.1. Hardware

The prototype was mostly implemented using standard machine parts along with additively manufactured parts and few machined parts. The high amount of standard parts has a significant impact on the reduction of the costs of such a machine. Furthermore, as an alternative to below-mentioned selected high-quality parts, such as ball screw linear modules, also basic components as known from low-cost machines like 3D printers may be used, depending on the particular requirements, such as load, respectively.

Six ball screw linear modules are arranged in parallel and attached to the additively manufactured frame. Each of them is driven by a locally closed-loop stepper motor with hall effect encoder. The actuators are driven by a modular 6-channel stepper driver mainboard which is often used in the field of 3D printing. The board performs the simultaneous interpolation of the axes and can be connected with various sensors to provide custom features like in-process overload protection.

The robot's rods are of fixed length and connected with the carriages through double row ball bearings and universal joints and with the platform through universal joints. The rods also include one-dimensional force transducers.

The platform is also 3D printed and can flexibly be extended with the help of its drill pattern.

The pitch of the ball screws has been selected to find a good compromise between power transmission and movement speed. Drive and gear efficiencies have been considered by all-inclusive safety margins.

Table 8.1 shows the robots' parameters for $0 \leq \alpha \leq 360°, 0° \leq \beta \leq 20°, |\gamma| \leq 20°$ and simulation resolution of $10mm$ and $20°$, respectively. The resulting workspace's bounding box is approximately $80mm \cdot 80mm \cdot 140mm$.

Table 1. Representative parameters obtained for the hexapod prototype

Symbol	Value	Unit
r_c	135	mm
α_c	30	°
r_p	65	mm
α_p	38	°
l_r	228	mm
l_g	200	mm
f	$[0, 0, 150]^T$	N
t	$[0, 0, 3]^T$	Nm
f_c	150	N
f_f	200	N
Amount	**Description**	
11,540	positions in the initial set after first exclusions	
1,314	orientations to check for each position	
15,163,560	poses evaluated geometrically	
106,315	geometrically valid poses	
102,462	valid poses under defined load	

8.2. Software implementation and real time capability

The robot prototype is designed to be able to interpret *GCODE* to establish the desired platform movement and communicates with a PC over USB. Over this interface it receives all control commands and also transmits status and measurement data.

A control software has been developed to define robotic movements and to generate the machine code which is then sent to the robot. Like this, complex 3D movements are expressed in the form of small linear segments. The robot itself then performs fine interpolation.

This makes use of the inverse kinematics included in the simulation. The forward kinematics is used to do in-process measurements mapping and validation, like forces, for instance.

9. Conclusion

A performant simulation driven development framework has been presented that allows for both the evaluation of existing hexapod robots as well as the design and dimensioning of custom variations and is characterized by the ease of use. Like this, application-related machines for precise laboratory handling tasks, machining or material forming, where high process forces are involved, can be realized quickly. It guides through the design, the analysis and the simulation and thus interactively supports the design process for non-experts, also. The simulation computes the entire workspace, but also allows for trajectory simulations and considers loads. Most parts of the novel simulation's logic are reused for the actual control of the developed robot. It has been validated by bringing a new design towards a working condition.

In the future, this tool will be further extended, by the traditional *Gough*/*Stewart*-platform for first, as well as by more detailed analyses for the kinematic's accuracy, its dynamics (including actuator dynamics) and further singularity insights. Using the derived inverse and direct kinematics, closed-loop robot control could be implemented in an embedded controller.

CRediT author statement

Lukas Bath: Conceptualization, Methodology, Software, Investigation, Writing - Original Draft, Writing - Review & Editing, Visualization. Thorsten Schüppstuhl: Writing - Original Draft, Supervision, Resources, Funding acquisition, Project administration.

Acknowledgements

Supported by:
Federal Ministry for Economic Affairs and Energy
on the basis of a decision by the German Bundestag

The work was performed in collaboration with Lufthansa Technik AG and funded by German Federal Ministry of Economic Affairs and Energy under grant number 20X1709B.

References

[1] Jean-Pierre Merlet. *Parallel robots*, volume 128. Springer Science & Business Media, 2005.

[2] Xin-Jun Liu and Jinsong Wang. Parallel kinematics. *Springer Tracts in Mechanical Engineering*, 2014.

[3] YD Patel, PM George, et al. Parallel manipulators applications—a survey. *Modern Mechanical Engineering*, 2(03):57, 2012.

[4] EF Fichter, DR Kerr, and J Rees-Jones. The gough—stewart platform parallel manipulator: A retrospective appreciation. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 223(1):243–281, 2009.

[5] Doug Stewart. A platform with six degrees of freedom. *Proceedings of the institution of mechanical engineers*, 180(1):371–386, 1965.

[6] Haiying Wen, Weiliang Xu, and Ming Cong. Kinematic model and analysis of an actuation redundant parallel robot with higher kinematic pairs for jaw movement. *IEEE Transactions on Industrial Electronics*, 62(3):1590–1598, 2014.

[7] Madusudanan Sathia Narayanan, Sourish Chakravarty, Hrishi Shah, and Venkat N Krovi. Kinematic-, static-and workspace analysis of a 6-p-us parallel manipulator. In *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, volume 44106, pages 1456–1456, 2010.

[8] Wang Chaoqun and Wu Hongtao. Singularity analysis of 6-pus vertical parallel mechanism [j]. *Key Engineering Materials*, 568:129–134, 2013.

[9] Daniel Lazard and J-P Merlet. The (true) stewart platform has 12 configurations. In *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*, pages 2160–2165. IEEE, 1994.

[10] Carlo Innocenti. Forward kinematics in polynomial form of the general stewart platform. *J. Mech. Des.*, 123(2):254–260, 2001.

[11] Tae-Young Lee and Jae-Kyung Shim. Algebraic elimination-based real-time forward kinematics of the 6-6 stewart platform with planar base and platform. In *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No. 01CH37164)*, volume 2, pages 1301–1306. IEEE, 2001.

[12] Charles C Nguyen, Sami C Antrazi, Zhen-Lei Zhou, and Charles E Campbell Jr. Analysis and implementation of a 6 dof stewart platform-based robotic wrist. *Computers & electrical engineering*, 17(3):191–203, 1991.

[13] Josep M Porta and Federico Thomas. Yet another approach to the gough-stewart platform forward kinematics. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 974–980. IEEE, 2018.

[14] Ana Lucia Cruz Ruiz, Stéphane Caro, Philippe Cardou, and François Guay. Arachnis: Analysis of robots actuated by cables with handy and neat interface software. In *Cable-Driven Parallel Robots*, pages 293–305. Springer, 2015.

[15] Metin Toz and Serdar Kucuk. Parallel manipulator software tool for design, analysis, and simulation of 195 gsp mechanisms. *Computer Applications in Engineering Education*, 23(6):931–946, 2015.

[16] I Bayati, Marco Belloli, Davide Ferrari, F Fossati, and Hermes Giberti. Design of a 6-dof robotic platform for wind tunnel tests of floating wind turbines. *Energy Procedia*, 53:313–323, 2014.

[17] Javier Ruiz-García, Daniel Chaparro-Altamirano, Ricardo Zavala-Yoe, and Ricardo Ramírez-Mendoza. Direct kinematics of a 6-pus parallel robot using a numeric-geometric method. In *2013 International Conference on Mechatronics, Electronics and Automotive Engineering*, pages 46–50. IEEE, 2013.

[18] R Zavala-Yoé, R Ramírez-Mendoza, and J Ruiz-García. Mechanical and computational design for control of a 6-pus parallel robot-based laser cutting machine. *Advances in Military Technology*, 10(1), 2015.

[19] Hamidreza Hajimirzaalian, Hasan Moosavi, and Mehdi Massah. Dynamics analysis and simulation of parallel robot stewart platform. In *2010 The 2nd International Conference on Computer and Automation Engineering (ICCAE)*, volume 5, pages 472–477. IEEE, 2010.

[20] Mohammad Heidar Khamsehei Fadaei, Seyed Ali Hamzeh Pahnehkolaei, Maryam Jafari Hesarlou, and Zahra Torkan. Dynamics modeling of a stewart platform in simulink msc adams. In *2017 IEEE 4th International Conference on Knowledge-Based Engineering and Innovation (KBEI)*, pages 0294–0298. IEEE, 2017.

[21] Shihui Zhang and Lingfu Kong. Analysis on kinematics and dynamics performance indices for 6-pus parallel robot mechanism. In *2007 IEEE International Conference on Networking, Sensing and Control*, pages 501–506. IEEE, 2007.

[22] WANG Hao CHEN Genliang LIN Zhongqin. Forward dynamics analysis of the 6-pus mechanism based on platform-legs composite simulation. *Chinese Journal of Mechanical Engineering*, 22(4):1, 2009.

[23] Yongsheng Zhao, Duan Yanbin, Liang Shunpan, Dou Yuchao, and Zeng Daxing. System debugging and experimental analysis of the 6-pus/upu parallel manipulator. In *2010 International Conference on Computer, Mechatronics, Control and Electronic Engineering*, volume 2, pages 376–381. IEEE, 2010.

[24] J-P Merlet and D Daney. Appropriate design of parallel manipulators. In *Smart Devices and machines for advanced manufacturing*, pages 1–25. Springer, 2008.