Parallel Mechanisms and Robots

Chapter · January 2008

DOI: 10.1007/978-3-540-30301-5_13

Jean-Pierre Merlet
National Institute for Research in Computer Science and Control Sophia-Antipolis

Clément Gosselin
Laval University

12. Parallel Mechanisms and Robots

Jean-Pierre Merlet, Clément Gosselin

This chapter presents an introduction to the kinematics and dynamics of parallel mechanisms, also referred to as parallel robots. As opposed to classical serial manipulators, the kinematic architecture of parallel robots includes *closed-loop kinematic chains*. As a consequence, their analysis differs considerably from that of their serial counterparts. This chapter aims to presenting the fundamental formulations and techniques used in their analysis.

12.1 Definitions

A *closed-loop kinematic chain* is one in which the links and joints are arranged such that at least one closed loop exists. Furthermore, a *complex closed-loop kinematic chain* is obtained when one of the links – other than the base – has a degree of connectivity greater than or equal to three, i. e., one link other than the base is connected through joints to at least three other links. A *parallel manipulator* can be defined as a closed-loop mechanism composed of an end-effector having n degrees of freedom and a fixed base, linked together by at least two independent kinematic chains.

An example of such a structure was patented in 1928 by *Gwinnett* [12.1] to be used as a platform for a movie theater. In 1947 *Gough* [12.2] established the basic principles of a mechanism with a closed-loop kinematic structure (Fig. 12.1) that allows the positioning and orientation of a moving platform so as to test tire wear and tear. He built a prototype of this machine in 1955. For this mechanism, the moving effector is a hexagonal platform whose vertices are all connected to a link by a ball-and-socket (spherical) joint. The other end of the link is attached to the base by a universal joint. A linear actuator allows the modification of the total length of the link; this mechanism is therefore a closed-loop kinematic structure, actuated by six linear actuators.

In 1965, *Stewart* [12.3] suggested the use of such a structure for flight simulators and the Gough mechanism is sometimes referred to as a *Stewart platform*. The same architecture was also concurrently proposed by *Kappel* as a motion simulation mechanism [12.4]. Nowadays the Gough platform is the platform of choice for flight simulators. The latter application provides a convincing illustration of the main advantage of parallel robots namely, their load-carrying capacity. Indeed, while the ratio of the mass of the payload over the mass of the robot is typically smaller than 0.15 for serial 6R industrial robots, this ratio can be larger than 10 for parallel structures. Another advantage of the Gough platform is

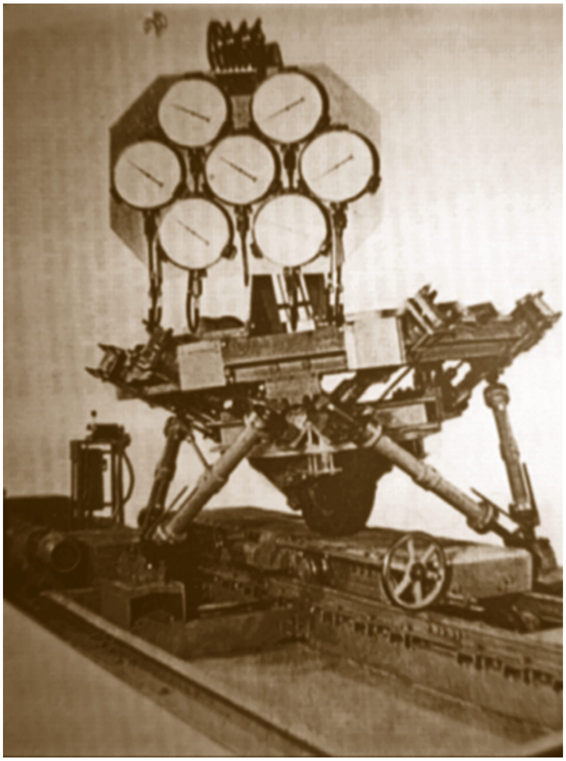

Fig. 12.1 Gough platform (1947)

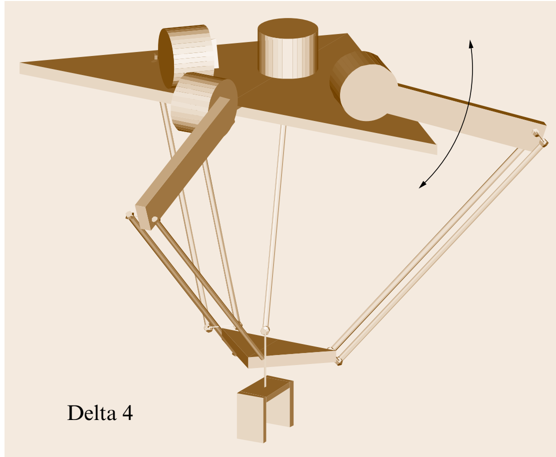

Fig. 12.2 The Delta robot

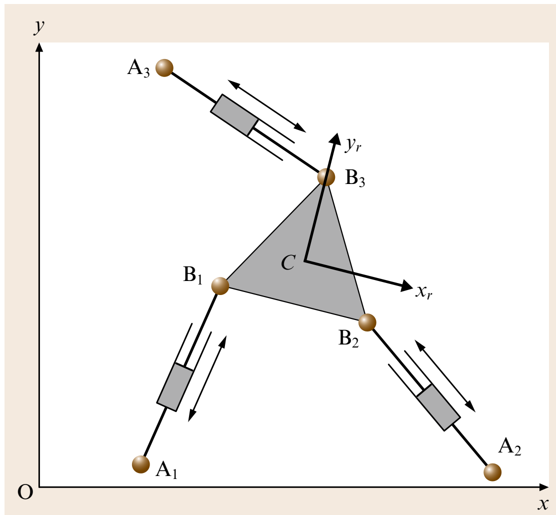

Fig. 12.3 The $3-R\underline{P}R$ planar parallel robot

its very good positioning accuracy. This high accuracy arises from the fact that the legs are working essentially in tension/compression and are subjected to virtually no bending – thereby leading to very small deformations – and from the fact that the errors in the internal sensors of the robot (the measurement of the lengths of the legs for the Gough platform) are mapped into very small errors of the platform position. Parallel robots are also almost insensitive to scaling (the same structure can be used for large or micro robots) and they can be built using almost any type of actuator or transmission, for example, wire transmissions can be used (see the *Robocrane* [12.5]). The main drawbacks of parallel robots are their small workspace and the singularities that can appear within the latter.

Apart from the Gough platform, the most successful designs of parallel robots are the Delta robot proposed by *Clavel* [12.6] (Fig. 12.2) and some planar parallel robots. The most common planar parallel robots have three identical legs each having an $R\underline{P}R$ or $\underline{R}RR$ architecture, where the underlined joint is the actuated one; such robots are often denoted as $3-R\underline{P}R$ (see Fig. 12.3) and $3-\underline{R}RR$.

The geometric arrangement of the joints and links of the Delta structure provides three translational degrees of freedom at the platform. Numerous other types of parallel robots have been proposed in recent years. Although most existing architectures are based on the intuition of their designer, the synthesis of parallel mechanisms can be dealt with systematically. An overview of the main approaches to the type synthesis of parallel mechanisms is given in the next section.

12.2 Type Synthesis of Parallel Mechanisms

Determining all potential mechanical architectures of parallel robots that can produce a given motion pattern at the moving platform is a challenging problem. Several authors have addressed this problem, referred to as type synthesis. The approaches proposed in the literature can be divided into three groups:

- *Approaches based on graph theory*: the enumeration of all possible structures having a given number of degrees of freedom (DOFs) can be performed by considering that there is only a finite set of possible kinematic pairs, and hence a very large, but finite, set of possible structure combinations (see, for instance, [12.7]). Classical mobility formulae – e.g., the Chebychev–Grübler–Kutzbach formula – are then used to determine the number of DOFs in the structure. Unfortunately these formulae do not take into account geometric properties of the mechanism that can change the number of DOFs of the platform. Therefore, synthesis approaches based strictly on graph theory can produce only limited results and have been largely superseded by the other two groups.
- *Approaches based on group theory*: the motion of a rigid body has the special structure of a group, the *displacement group*. Subgroups of the group of displacements, such as the spatial translations or all the translational and rotational motions about all axes that are parallel to the axis defined by a given vector (Schönflies motion) play an important role as they can be combined through the *intersection* operation [12.8] when elements of subgroups act on the same rigid body. Type synthesis consists in determining all the possible subgroups to which the different kinematic chains that will constitute the legs of the robot may belong so that their intersection leads to the desired motion pattern for the platform. Synthesis approaches based on group theory led to the discovery of numerous possible architectures. Nevertheless, the displacement group has special properties that are not reflected by its group structure alone. Also, the approach is limited to motion patterns that can be described by a subgroup of the displacement group.
- *Approaches based on screw theory*: in this approach, the first step is to determine the wrench system S that is reciprocal to the desired velocity twist of the moving platform. Then, the wrenches of the kinematic chains of the robot whose union spans the system S (and that determine all the possible structures of the kinematic chains that will generate the corresponding wrenches) are enumerated. Then, since all considered twists and wrenches are instantaneous, it is necessary to verify that the mobility of the platform is full-cycle and not only instantaneous. A systematic implementation of this approach is provided in [12.9].

These synthesis methods have been used to generate a large number of architectures that cannot be presented in this book, but the Web site of Merlet [12.10] presents a comprehensive description of a large number of mechanical architectures. Among others, several remarkable architectures were proposed in the recent years to fully or partially decouple translational motions or Schönflies motions (see [12.9] and the references therein).

It should also be pointed out that for robots with fewer than six DOFs, synthesis methods generally lead to architectures that must satisfy stringent geometric constraints (such as for the Delta robot, where the axes of the parallelograms should be parallel and their sides of equal length). Such constraints cannot be exactly verified in practice and consequently the manipulator will exhibit *parasitic*, i. e., unwanted, motion. Open issues therefore include:

- the determination of the maximal amplitude of the parasitic motion over a given workspace and for a given robot
- the determination of the geometry of a given structure such that the maximal amplitude of the parasitic motion will be smaller than a given threshold

12.3 Kinematics

12.3.1 Inverse Kinematics

The solution of the inverse kinematic problem is usually straightforward for parallel robots. This can be illustrated with the Gough platform. The solution of the inverse kinematic problem consists in determining the vector of leg lengths $\boldsymbol{q}$ for a given pose of the platform defined by the position vector $\boldsymbol{p}$ of a given point on the platform in a fixed reference frame and a rotation matrix $\boldsymbol{R}$ representing the orientation of the platform with respect to the fixed frame. Let $\boldsymbol{a}_i$ denote the position vector of the anchor point of the ith leg on the base given in the fixed reference frame and $\boldsymbol{b}_i$ denote the position vector of the anchor point of the ith leg on the platform, given in a moving frame attached to the platform. The length of the ith leg is the norm of the vector connecting the two anchor points. The latter vector, denoted by $\boldsymbol{s}_i$, can be written as

$$\boldsymbol{s}_i = \boldsymbol{p} + \boldsymbol{R}\boldsymbol{b}_i - \boldsymbol{a}_i \,, \quad i = 1, \ldots, 6 \,. \tag{12.1}$$

Given the pose of the platform (vector $\boldsymbol{p}$ and matrix $\boldsymbol{R}$), the vector $\boldsymbol{s}_i$ is readily calculated using (12.1), and hence the leg lengths can be obtained.

12.3.2 Forward Kinematics

The solution of the forward kinematic problem consists in determining the pose of the platform for a given set of actuated joint coordinates (a given vector $\boldsymbol{q}$). This solution is needed for control purposes, calibration, and motion planning.

The forward kinematic problem of a parallel robot is usually much more complex than the inverse kinematic problem. Indeed, the loop closure equations (12.1) are typically highly nonlinear expressions of the pose variables. They form a nonlinear set of equations that generally admits multiple solutions (for example the Gough platform can have up to 40 solutions [12.11–13], while a table with the number of solutions for Gough platforms with special geometries is provided in [12.14]). The forward kinematic problem arises in two different contexts. In the first case, no estimate of the current pose of the platform is available (for example, when starting the robot) while in the second case, a relatively precise estimate of the pose is known (for example in real-time control when the forward kinematics has been solved at the previous sampling time of the controller). In the first case, the only known approach is to determine all the solutions of the inverse kinematic equations, although there is no known algorithm to sort the set of solutions. It is often possible to determine an upper bound on the number of real solutions. Consider for example the case of the planar $3-R\underline{P}R$ robot (Fig. 12.3). If the joint at point B_3 is disassembled, two separate mechanisms are obtained, namely a four-bar linkage and a rotary lever. From the kinematics of four-bar linkages it is known that point B_3 moves on a sixth-order algebraic curve. Moreover, it is also known that point B_3 of the lever moves on a circle, i. e., a second-order algebraic curve. For a given set of actuated joint coordinates, solutions to the forward kinematic problem arise when the two curves intersect, i. e., when the mechanism can be assembled. Bezout's theorem states that algebraic curves of order m, n intersect at nm points, counting their order of multiplicity. In the above case, this means that the curves intersect in 12 points. However, these 12 points include the two circular imaginary points that belong to the coupler curve of the four-bar linkage and to any circle, thereby to their intersection. These points are counted three times in Bezout's theorem and hence the forward kinematic problem will have at most six real solutions, corresponding to the intersection points.

Various methods have been proposed for the solution of the forward kinematic problem: elimination [12.15], continuation [12.13], Gröbner bases [12.16], and interval analysis [12.17]. Elimination is usually not very stable numerically (i. e., it can produce spurious roots and miss solutions) unless special care is taken in the implementation of the resulting univariate equation and the elimination steps. For instance, the transformation of polynomial solutions into eigenvalue problems can be used [12.18]. Polynomial continuation, on the other hand, is much more stable numerically since mature algorithms can be found in the literature [12.19]. The fastest methods – although not appropriate for real-time use – are Gröbner bases and interval analysis. They also have the advantage of being numerically certified (no roots can be missed and the solution can be computed with an arbitrarily prescribed accuracy).

However, in the simplest cases, elimination can usually be used to produce stable implementations. Consider, for example, the case of the planar $3-R\underline{P}R$ robot (Fig. 12.3), where the fixed reference frame has its origin at point A_1 and its x-axis passes through the point A_2. Similarly, point B_1 is chosen as the origin of the moving frame while the x-axis of this frame passes through point B_2. The pose of the platform is then defined by the coordinates (x, y) of point B_1 in the fixed reference frame and by the angle θ between the x-axes of the reference and moving frames. The known length q_1 of link 1 can thus be written simply as

$$q_1^2 = x^2 + y^2 \,. \tag{12.2}$$

Then, the length q_i of the other links can be written as:

$$q_i^2 = x^2 + y^2 + g_i(x, y, \theta) \,, \quad i = 2, 3 \,, \tag{12.3}$$

where g_i is linear in x, y. Subtracting the length of link 1 from the latter two equations leads to a linear system of two equations in the unknowns x and y. This system is solved and x and y are obtained as functions of θ. The expressions for x and y are then substituted into (12.2), thereby leading to a single equation in the unknown θ. This unknown appears in the equation through its sine and cosine, and by using the Weierstrass substitution, the equation can be transformed into a sixth-order polynomial equation in the unknown $T = \tan(\theta/2)$. Solving this equation in T allows one to compute all possible solutions in θ, from which the values of x and y are readily obtained. It is noted that this polynomial is of minimal degree since it was established that the number of real solutions cannot exceed six. It can also be shown that there exist configurations of the robot for which there are exactly six real solutions to the forward kinematics. Finally, it should be pointed out that the forward kinematic problem of the $3-\underline{R}RR$ robot is mathematically exactly the same as the one developed above for the $3-R\underline{P}R$ robot. This is typical for parallel robots.

When a priori information (an initial guess of the solution) is available, the forward kinematics is usually solved using the Newton–Raphson or the Newton–Gauss iterative scheme. Consider the solution to the inverse kinematic problem, written as:

$$\boldsymbol{q} = \boldsymbol{f}(\boldsymbol{x}) \, . \tag{12.4}$$

Iteration k of the Newton–Raphson procedure is written as

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \boldsymbol{A}(\boldsymbol{q} - \boldsymbol{f}(\boldsymbol{x}_k)) \, , \tag{12.5}$$

where $\boldsymbol{q}$ is the vector of prescribed joint variables. The matrix $\boldsymbol{A}$ is usually chosen as $(\frac{\partial \boldsymbol{f}}{\partial q})^{-1}(\boldsymbol{x}_k)$ (the inverse matrix may not be computed at each iteration or even be chosen as constant). The iterative scheme stops when the magnitude of the difference vector $(\boldsymbol{q} - \boldsymbol{f}(\boldsymbol{x}_k))$ is smaller than a chosen threshold.

The choice of the inverse kinematics equations plays an important role in the convergence of this method [12.20]. For example, for a Gough platform, a minimal set of equations (six equations involving six unknowns: three for the translation and three orientation angles) can be used, but other representations are possible. For instance, the position coordinates of three of the platform anchor points – in the fixed reference frame – can be used as unknowns (the coordinates of the remaining three points are then readily computed upon convergence). Using this representation, nine equations are needed. Six equations are obtained based on the known distances between the anchor points on the base and platform and three additional equations are obtained based on the known distances between all combinations of the three selected platform anchor points.

Provided that a good initial estimate of the solution is available, the Newton–Raphson algorithm is usually very fast. However, the procedure may not converge or, even worse, it may converge to a solution that is not the correct pose of the platform, i. e., it may converge to another assembly mode. Such a situation may occur even if the initial guess is arbitrarily close to the correct pose. If the result is used in a control loop, the consequences can be catastrophic. Fortunately mathematical tools such as the Kantorovitch theorem combined with interval analysis can be used to determine if the solution found by the Newton–Raphson scheme is the correct pose of the robot, i. e., to certify the result at the cost of a larger computation time, yet still compatible with real time [12.17].

Another possible approach for the solution of the forward kinematic problem is either to add sensors on the passive joints (e.g., on the U joints of a Gough platform) or to add passive legs with sensed joints. The main issue is then to determine the number and location of the additional sensors that will lead to a unique solution [12.21, 22] and to determine the effect of the sensor errors on the positioning error of the platform. For example, Stoughton mentions that, for a Gough platform with sensors in the U joints, it was still necessary to use the Newton–Raphson scheme to improve the accuracy of the solution because the solutions obtained with the additional sensors were very sensitive to measurement noise [12.23].

12.4 Velocity and Accuracy Analysis

Similarly to serial robots, the actuated joint velocity vector of parallel robots $\dot{\boldsymbol{q}}$ is related linearly to the vector of translational and angular velocities of the platform (for simplicity, the latter vector is denoted here by $\dot{\boldsymbol{p}}$, although the angular velocity is not the time derivative of any set of angular parameters). The linear mapping between the two vectors is provided by the Jacobian matrix $\boldsymbol{J}$:

$$\dot{\boldsymbol{p}} = \boldsymbol{J}(\boldsymbol{p})\dot{\boldsymbol{q}} \, . \tag{12.6}$$

However, for parallel robots, a closed-form expression is usually available for the inverse Jacobian $\boldsymbol{J}^{-1}$, but this is much more difficult for $\boldsymbol{J}$ (more precisely, the closed form for most six-DOF robots will be so complex that it cannot be used in practice). For example, a simple static analysis can be used to show that the ith row $\boldsymbol{J}_i^{-1}$ of matrix $\boldsymbol{J}^{-1}$ for a Gough platform can be written as

$$\boldsymbol{J}_i^{-1} = \boldsymbol{n}_i^\top \;\; (\boldsymbol{c}_i \times \boldsymbol{n}_i)^\top \;, \tag{12.7}$$

where $\boldsymbol{n}_i$ represents the unit vector defined along leg i and $\boldsymbol{c}_i$ is the vector connecting the origin of the mobile frame attached to the platform to the ith anchor point on the platform.

The effect of the joint sensor errors $\Delta\boldsymbol{q}$ on the positioning error $\Delta\boldsymbol{p}$ follows the same relationship, namely

$$\Delta\boldsymbol{p} = \boldsymbol{J}(\boldsymbol{p})\Delta\boldsymbol{q} \;. \tag{12.8}$$

Since the matrix $\boldsymbol{J}$ is difficult to obtain in closed form, accuracy analysis (i. e., finding the maximal positioning errors over a given workspace for bounded joint sensor errors) is much more difficult than for serial robots [12.20, 24]. Apart from the measurement errors, there are other sources of inaccuracy in parallel robots, namely: clearance in the passive joints, manufacturing tolerances, thermal errors, and gravity-induced and dynamic errors [12.25, 26]. The effect of joint clearances on trajectories followed by serial and parallel robots was studied in [12.27–29]. According to these works, it is impossible to determine trends for the effect of the geometric errors: a case-by-case study must be performed since the effect is highly dependent on the mechanical architecture, dimensioning, and workspace of the robot. Thermal effects are sometimes mentioned as possible sources of inaccuracy, although few works substantiate this claim [12.30] and cooling may slightly reduce their effects [12.31].

Calibration is another means of improving the accuracy of parallel robots. This issue was addressed in Chap. 3. The methods and procedures used for parallel robots differ slightly from the ones used for serial robots since only the inverse kinematic equations are available and since the positioning of the platform is much less sensitive to the geometric errors than in serial robots [12.32, 33]. Hence the measurement noise occurring during calibration has a significant impact and may lead to surprising results. For example, the classical least-squares method may lead to parameters that are such that some constraint equations are not satisfied even when the measurement noise is taken into account [12.34]. It has also been shown with experimental data that classical parallel robot modeling leads to constraint equations that do not have any solution irrespective of the measurement noise [12.35]. Moreover, calibration is very sensitive to the choice of the calibration poses [12.36]: it seems that the optimal choice is to select poses on the workspace boundary [12.37, 38].

12.5 Singularity Analysis

12.5.1 General Formulation

The analysis of singularities of parallel mechanisms was first addressed by *Gosselin* and *Angeles* [12.39]. In this formulation, the kinematic equations were reduced to the input–output relationship between the actuated joint coordinate vector $\boldsymbol{q}$ and the platform Cartesian coordinate vector $\boldsymbol{p}$, namely

$$\boldsymbol{f}(\boldsymbol{q}, \boldsymbol{p}) = 0 \;. \tag{12.9}$$

Differentiating (12.9) with respect to time leads to

$$\boldsymbol{B}\dot{\boldsymbol{q}} + \boldsymbol{A}\dot{\boldsymbol{p}} = \boldsymbol{0} \;. \tag{12.10}$$

Three types of singularities can then be defined:

- when matrix $\boldsymbol{B}$ is singular (termed serial singularity)
- when matrix $\boldsymbol{A}$ is singular (termed parallel singularity)
- when the input–output equations degenerate (termed an architecture singularity) in which case matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ can both be singular.

In a serial singular configuration, the joints can have a nonzero velocity while the platform is at rest. In a parallel singularity, there exist nonzero platform velocities for which the joint velocities are zero. In the neighborhood of such a configuration, the robot can undergo an infinitesimal motion while the actuators are locked. Since the mobility of the end-effector should be zero when the actuators are locked, it is also said that in such a configuration the robot *gains* some DOFs. As a consequence, certain degrees of freedom of the platform cannot be controlled, and this is a major problem.

A more general study of singularities was proposed by *Zlatanov* [12.40]. The latter analysis uses the velocity equations involving the full twist of the end-effector and all joint velocities (passive or actuated). This approach led to a more detailed classification of singularities and was also used later to reveal special singularities (referred to as *constraint singularities*) [12.41] that could not be found with the analysis presented in [12.39].

The singularity analyses discussed above are of the first order. Second- (and higher-)order singularity analyses can also be performed, although the latter are much more complex [12.29, 42].

12.5.2 Parallel Singularity Analysis

This type of singularity is especially important for parallel robots because it corresponds to configurations in which the control of the robot is lost. Furthermore, very large joint forces can occur in the vicinity of singular poses, that may lead to a breakdown of the robot. The main issues to be addressed in this context are:

- the characterization of the singularities
- the definition of performance indices representing the closeness to a singularity
- the development of algorithms that are capable of detecting the presence of singularities over a given workspace or trajectory

Parallel singularities arise when the 6×6 *full inverse Jacobian* $\boldsymbol{J}_f$ (i. e., the matrix that maps the full twist of the platform into active – and eventually passive – joint velocities) is singular, i. e., when its determinant $\det(\boldsymbol{J}_f)$ is equal to 0. It is pointed out that passive joint velocities sometimes have to be included because restricting the analysis to the active joint velocities may not allow the determination of all singular configurations of the mechanism (see the example of the $3-U\underline{P}U$ robot [12.43]). Usually, a proper velocity analysis allows one to establish this matrix in closed form, but computing the determinant of this matrix may be difficult even with symbolic computation tools (see the example of the Gough platform [12.44, 45]). The advantage of this approach is that once the expression of the determinant is obtained, the locus of singular configurations can be plotted in the workspace, thereby leading to graphical representations that can be useful in a context of design. However, the determinant itself is usually a large expression that does not provide insight into the geometric conditions associated with the singularities.

An alternative approach is to use line geometry: indeed, for several parallel robots (although not all), a row of $\boldsymbol{J}_f$ corresponds to the Plücker vector of some line defined on the links of the robot. For example, for a Gough platform, the rows of $\boldsymbol{J}_f$ are the normalized Plücker vectors of the lines associated with the legs of the robot. A singularity of matrix $\boldsymbol{J}_f$ therefore implies a linear dependency between these vectors (they then constitute a *linear complex*), a situation that may occur only if the lines associated with the vectors verify particular geometric constraints [12.46] (for example, three Plücker vectors will be linearly dependent if and only if their associated lines are coplanar and intersect at a common point). These geometric constraints have been identified by Grassmann for every set of three, four, five or six vectors. Singularity analysis is thus reduced to determining conditions on the pose parameters for which these constraints are satisfied, giving geometric information on the singularity variety. The closed-form singularity condition can then be substituted into the Jacobian matrix and the kernel of the matrix calculated in order to determine the singular motion [12.47].

Measuring closeness between a pose and a singular configuration is a difficult problem: there exists no mathematical metric defining the distance between a prescribed pose and a given singular pose. Hence, a certain level of arbitrariness must be accepted in the definition of the *distance* to a singularity and none of the proposed indices is perfect. For example, a possible index is the determinant of $\boldsymbol{J}_f$: unfortunately when the platform is subjected to both translational and rotational motion, the latter matrix is not dimensionally homogeneous and hence the value of the determinant will change according to the physical units used to describe the geometry of the robot. Dexterity indices, as defined in Chap. 11 (although they are less relevant for parallel robots than for serial robots [12.24]), can also be used and special indices for parallel robots have also been proposed [12.48, 49].

Most of the analyses based on the above indices are local, i. e., valid only for a given pose, while in practice the problem is to determine if a singular configuration can occur over a given workspace or trajectory. Fortunately, an algorithm exists that allows this verification, even if the geometric model of the robot is uncertain [12.50]. However, it should be pointed out that a singularity-free workspace may not always be optimal for parallel robots. Indeed, other performance requirements may impose the presence of singularities in the workspace or, the robot may have part of its workspace singularity-free (for example its working region) while exhibiting singularities outside this region. Therefore, a motion planner producing a trajectory that avoids singularities while remaining close to a desired path is advisable and various approaches have been proposed to address this problem [12.51, 52].

Finally, it is noted that being close to a singular configuration may be useful in some cases. For example, large amplification factors between the end-effector motion and the actuated joint motion may be interesting for fine-positioning devices with a very small workspace or for improving the sensitivity along some measurement directions for a parallel robot used as a force sensor [12.53]. It should also be mentioned that parallel robots that remain *permanently* in a singular configuration may be interesting since they are capable of producing complex motions with only one actuator [12.54–56].

12.6 Workspace Analysis

As mentioned in the introduction, one of the main drawbacks of parallel robots is their small workspace. Moreover, their workspace analysis is usually more complex than for serial robots, especially if the platform has more than 3 DOFs. Indeed, the platform motion capabilities are usually coupled, thereby prohibiting simple graphical representations of the workspace. *Decoupled robots* have also been proposed [12.57–59] but they do not provide the load-carrying capacity of conventional parallel robots. In general, the workspace of parallel robots is restricted by

- the limitations on the actuated joint variables: for example, the length of the legs of a Gough platform have a minimum and maximum value,
- the limitations on the range of motion of the passive joints: for example the U and S joints of a Gough platform have restricted ranges of angular motion,
- the mechanical interferences between the bodies of the mechanism (legs, base, and platform).

Different types of workspaces can be defined such as the *dextrous* workspace (the set of platform locations in which all rotations are possible), the *maximal* workspace (the set of platform locations that may be reached with at least one orientation) or the *orientation* workspace (the set of orientations that can be reached for a given location of the platform).

A possible approach for representing the workspace of parallel robots with $n > 3$ DOFs is to fix the value of $n-3$ pose parameters in order to be able to plot the workspace over the remaining three DOFs. Such plots can be obtained very efficiently using geometric methods if the plotted variables involve only translational motions because a geometric approach usually allows one to establish the nature of the boundary of the workspace (see, for example, [12.60, 61] for the calculation of the workspace of a Gough platform when its orientation is fixed). Another advantage of this approach is that it allows the computation of the surface and volume of the workspace while being very efficient in terms of storage space. However, when rotational motion is included, the geometric approach becomes rather complex. Possible alternatives include:

- discretization methods in which all poses of a n-dimensional grid are checked with respect to the kinematic constraints. Such methods are computationally intensive because the computation time increases exponentially with the resolution of the mesh. Also they require a large storage space. On the other hand, the advantage of discretization is that it is usually simple to implement and it allows one to take into account all kinematic constraints as they can usually be simply verified for a given pose,
- numerical methods that allow the determination of the workspace boundary [12.62, 63],
- numerical methods based on interval analysis that allow the determination of an approximation of the workspace volume up to an arbitrary accuracy [12.64]. This representation is also appropriate for motion planning problems.

Singularities can also split the workspace calculated from the above kinematic constraints into elementary components, called *aspects* by *Wenger* [12.65] and which are separated by a singularity variety. However, determining the aspects for spatial robots is still an open problem. A parallel robot may not always be able to move from one aspect to another (at least without considering the dynamics of the robot [12.66]) and hence the useful workspace can be reduced.

A problem related to workspace analysis is the *motion planning problem*, which is slightly different from that encountered with serial robots. For parallel robots, the problem is not to avoid obstacles within the workspace but to determine if a given trajectory lies entirely inside the workspace of the robot or to determine a trajectory between two poses that lies in the workspace and is singularity-free. Algorithms for checking if a trajectory is valid are available [12.67] but finding such a trajectory is more difficult. Indeed, classical serial robot motion planners work in the joint space and assume that there is a one-to-one relationship between the joint space and the operational space. With this assumption, it is possible to determine a set of points in the joint space that is collision-free and to use this knowledge to build a collision-free path between two poses. However, this assumption is not valid for parallel robots because the mapping between the joint space and the operational space is not one-to-one: a point in the joint space may either correspond to multiple points in the operational space or have no correspondence because the closure equations of the mechanism are not satisfied. For parallel robots, the most efficient motion planner seems to be an adaptation of a probabilistic motion planner that takes into account – to a certain extent – the closure equations [12.68, 69].

Another motion planning problem is related to tasks where a limited number of DOFs of the robot are used. For example, when a six-DOF parallel robot is used for machining, the rotation of the platform around the axis of the tool is not used since the spindle rotation ensures the needed motion. Thus, an additional DOF is available that can be used to increase the machine's workspace, to avoid singularities or to optimize some performance index of the robot [12.70, 71]. Along the same lines, it is possible to define the *part positioning* problem [12.72] as determining the pose of a rigid body with respect to the workspace of the robot so that this pose satisfies some constraints (e.g., the rigid body should lie fully inside the workspace and at all points of its surface the robot should have some rotational ability).

12.7 Static Analysis and Static Balancing

Similarly to serial robots, the static analysis of parallel robots can be readily performed using the Jacobian matrix. Indeed, the mapping between the vector of actuated joint forces/torques $\boldsymbol{\tau}$ and the external wrench $\boldsymbol{f}$ exerted by the platform can be written as

$$\boldsymbol{\tau} = \boldsymbol{J}^\top \boldsymbol{f} \,, \tag{12.11}$$

where $\boldsymbol{J}^\top$ is the transpose of the Jacobian matrix of the robot. Equation (12.11) can be used for various purposes, namely

- during the *design* process in order to determine the actuator forces/torques (for actuator selection). In this case, the designer is interested in finding the maximal actuator forces/torques over the robot workspace. This is a complex issue since $\boldsymbol{J}^\top$ is not known in closed form,
- in applications where the robot is used as a *force sensor*: if $\boldsymbol{\tau}$ is measured and the pose of the platform is known, then (12.11) can be used to calculate $\boldsymbol{f}$ so that the robot can be used both as a motion and sensing platform [12.73–75].

The stiffness matrix $\boldsymbol{K}$ of a parallel robot is defined similarly to that of serial robots as

$$\boldsymbol{K} = \boldsymbol{J}^{-\top} \boldsymbol{K}_j \boldsymbol{J}^{-1} \,, \tag{12.12}$$

where $\boldsymbol{K}_j$ is the diagonal matrix of actuated joint stiffness. *Duffy* [12.76] notes that this derivation is, in general, incomplete. For example, for a Gough platform this derivation assumes that there is no initial load on the elastic element of the link. Assuming that the length of the unloaded link is q_i^0 leads to

$$\Delta \boldsymbol{f} = \sum_{i=1}^{i=6} k \Delta q_i \boldsymbol{n}_i + k_i (q_i - q_i^0) \Delta \boldsymbol{n}_i \,,$$

$$\Delta \boldsymbol{m} = \sum_{i=1}^{i=6} k \Delta q_i \boldsymbol{c}_i \times \boldsymbol{n}_i + k_i (q_i - q_i^0) \Delta (\boldsymbol{c}_i \times \boldsymbol{n}_i) \,,$$

where k_i denotes the axial stiffness of the leg, $\boldsymbol{n}_i$ represents the unit vector defined along the ith leg, $\boldsymbol{c}_i$ is the vector connecting the origin of the moving reference frame to the ith anchor point of the platform, and $\boldsymbol{f}$ and $\boldsymbol{m}$ are the external force and moment vectors that are applied by the platform. Consequently, the stiffness matrix as defined in (12.12) is valid only if $q_i = q_i^0$, and is coined the *passive stiffness*.

Moreover, it was pointed out in the literature [12.77] that the formulation of (12.12) is valid only when the external wrench is zero. Indeed, the above formulation does not account for the fact that the Jacobian matrix is configuration dependent and therefore changes with the applied loads. The formulation proposed in [12.77] and termed the *conservative congruence transformation* (CCT) takes these changes into account and leads to a stiffness matrix defined as

$$\boldsymbol{K}_c = \boldsymbol{J}^{-\top} \boldsymbol{K}_j \boldsymbol{J}^{-1} + \left(\frac{\partial \boldsymbol{J}^{-\top} \boldsymbol{\tau}}{\partial \boldsymbol{p}} \right) \,, \tag{12.13}$$

where $\boldsymbol{p}$ is the vector of Cartesian coordinates and $\boldsymbol{\tau}$ is the vector of actuated joint forces/torques. Whenever possible, the formulation of (12.13) should be used instead of that of (12.12) because it is mechanically consistent.

Another interesting static problem is the static balancing of parallel robots. The static balancing of mechanisms in general has been an important research topic for several decades (see, for instance, [12.78] for an account of the state of the art and many recent new results). Parallel mechanisms are said to be *statically balanced* when the weight of the links does not produce any torque (or force) at the actuators under static conditions, for any configuration of the manipulator or mechanism. This condition is also referred to as *gravity compensation*. The gravity compensation of parallel mechanisms was addressed by *Dunlop* [12.79] who suggested the use of counterweights to balance a two-DOF parallel robot used for antenna aiming, and by *Jean* [12.80] for planar robots. The latter case leads to simple and elegant balancing conditions.

In general, static balancing can be achieved using counterweights and/or springs. When springs are used, static balancing can be defined as the set of conditions for which the total potential energy in the mechanism – including gravitational energy and the elastic energy stored in the springs – is constant for any configuration of the mechanism. When no springs – or other means of storing elastic energy – are used, then static balancing conditions imply that the center of mass of the mechanism does not move in the direction of the gravity vector, for any motion of the mechanism.

Consider a general spatial n-degree-of-freedom parallel mechanism composed of n_b moving bodies and one fixed link. Moreover, let the position vector of the center of mass of each moving body with respect to a fixed reference frame be noted $\boldsymbol{c}_i$ and let the mass of the ith moving body be noted m_i. The position vector of the center of mass of the mechanism with respect to the fixed frame, noted $\boldsymbol{c}$, can be written as

$$\boldsymbol{c} = \frac{1}{M} \sum_{i=1}^{n_\mathrm{b}} m_i \boldsymbol{c}_i \;, \qquad (12.14)$$

where M is the total mass of the moving links, i. e.,

$$M = \sum_{i=1}^{n_\mathrm{b}} m_i \;. \qquad (12.15)$$

In general, the vector $\boldsymbol{c}$ is a function of the configuration of the mechanism, i. e.,

$$\boldsymbol{c} = \boldsymbol{c}(\boldsymbol{\theta}) \;, \qquad (12.16)$$

where $\boldsymbol{\theta}$ is the vector comprising all the joint coordinates of the mechanism.

Following this notation, and if no elastic components are used, the condition for static balancing can be written as

$$\boldsymbol{e}_z^\top \boldsymbol{c} = C_\mathrm{t} \;, \qquad (12.17)$$

where C_t is an arbitrary constant and $\boldsymbol{e}_z$ is a unit vector oriented in the direction of gravity.

When elastic components are used, the total potential energy in the mechanism, denoted by V, is defined as the sum of the gravitational and elastic potential energy and can be written as

$$V = g\boldsymbol{e}_z^\top \sum_{i=1}^{n_\mathrm{b}} m_i \boldsymbol{c}_i + \frac{1}{2} \sum_{j=1}^{n_\mathrm{s}} k_j (s_j - s_j^0)^2 \;, \qquad (12.18)$$

where g is the magnitude of the gravitational acceleration, n_s is the number of linear elastic components in the system, k_j is the stiffness of the jth elastic component, s_j is the length of the j-th elastic component, and s_j^0 is its undeformed length. As mentioned above, when elastic elements are used, the condition for static balancing is that the total potential energy is constant, which can be written as

$$V = V_\mathrm{c} \;, \qquad (12.19)$$

where V_c is an arbitrary constant.

The general equations given above can be used to obtain the conditions under which a given mechanism is statically balanced. In general, the equations provide sufficient conditions for balancing. The problem of statically balancing spatial parallel robots is usually complex. Several papers were published recently on this subject [12.81, 82]. Among other results, it was pointed out that it is not possible to balance a $6-UPS$ robot in all poses by using only counterweights [12.83]. Alternative mechanisms (using parallelograms) that can be balanced using only springs were suggested [12.81, 84, 85].

A natural extension of static balancing is dynamic balancing. The dynamic balancing problem consists in finding parallel mechanisms that do not produce any reaction force or torque on their base for arbitrary trajectories. This problem was addressed for planar and spatial parallel mechanisms. In [12.86], it was shown that dynamically balanced parallel mechanisms with up to six DOFs can be synthesized, although the resulting mechanical architectures may be complex.

12.8 Dynamic Analysis

The dynamic model of parallel robots has a form similar to that of serial robots (see Chap. 2), namely

$$\boldsymbol{M}(\boldsymbol{x})\ddot{\boldsymbol{p}}+\boldsymbol{C}(\dot{\boldsymbol{p}},\dot{\boldsymbol{q}},\boldsymbol{p},\boldsymbol{q})+\boldsymbol{G}(\boldsymbol{p},\boldsymbol{q})=\boldsymbol{\tau}\ , \tag{12.20}$$

where $\boldsymbol{M}$ is the positive-definite generalized inertia matrix, $\boldsymbol{G}$ is the gravitational term, and $\boldsymbol{C}$ is the centrifugal and Coriolis term. However, this equation is more difficult to obtain for parallel robots because the closure equations have to be satisfied. A classical method for calculating the dynamic models of closed-chain mechanisms is to consider first an equivalent tree-structure, and then to enforce the kinematic constraints using Lagrange multipliers or d'Alembert's principle [12.87, 88], a principle that has proved to be valid in [12.89]. Other approaches include the use of the principle of virtual work [12.90–93], the Lagrange formalism [12.94–96], Hamilton's principle [12.97], and Newton–Euler equations [12.98–103].

The main drawbacks of dynamic models obtained for parallel robots is that they are usually rather complex, they require the determination of dynamic parameters that are often not well known, and they involve solving the forward kinematics. Therefore, their calculation is computer intensive while they must be used in real time.

Using dynamic models for control purposes was proposed [12.104, 105], usually in the context of an adaptive control scheme, in which the tracking errors are used online to correct the parameters used in the dynamic equations [12.106, 107]. Control laws were proposed mainly for planar robots and for the Delta robot [12.108–110] although some implementations have been proposed for general six-DOF robots [12.111, 112] or vibration platforms [12.113]. However for six-DOF robots, the benefits of using dynamic models for high-speed motions are difficult to establish because the computational burden of the dynamic model somewhat reduces the potential gains.

This is clearly an important and open issue since parallel robots can operate at velocities and accelerations much larger than those of serial robots. For example, some Delta robots have reached accelerations of the order of 500 $\mathrm{m/s^2}$ while wire-driven robots can probably exceed this value.

Finally, the optimization of the dynamic properties of parallel mechanisms can be addressed by minimizing the variation of the inertia over the workspace [12.114].

12.9 Design

The design of a robot can be decomposed into two main phases:

- *structural synthesis*: finding the general mechanical structure of the robot
- *geometric synthesis*: determining the value of the geometric parameters involved in a given structure (here geometric parameters must be understood in a loose sense, for example, mass and inertia may also be involved)

The problem of structural synthesis (type synthesis) was addressed in Sect. 12.2. However, performance requirements other than the motion pattern of the robot have to be taken into account in the design of a robot. Serial robots have the advantage that the number of possible mechanical architectures is relatively small and that some of these architectures have clear advantages in terms of performance compared to others (for example, the workspace of a $3R$ structure is much larger than the workspace of a Cartesian robot of similar size).

Unfortunately no such rules exist for parallel robots, for which there are furthermore a large number of possible mechanical designs. Additionally, the performances of parallel robots are very sensitive to their geometric parameters. For example, the extremal stiffness of a Gough platform over a given workspace can change by 700% for a change of only 10% of the platform radius. Consequently, structural synthesis cannot be dissociated from the geometric synthesis. In fact, it is conjectured that a well-dimensioned robot of any structural type will in general perform better than a poorly designed robot with a structure that may seem more appropriate for the task at hand.

Usually the design process is treated as an optimization problem. To each specified performance requirement is associated a performance index whose value increases with the level of violation of the performance requirement. These performance indices are summed in a weighted real-valued function called the *cost function*, which is a function of the geometric design parameters and then a numerical optimization procedure is used to find the parameters that minimize the cost function (hence this approach leads to what is called an *optimal design*) [12.115–117]. There are however numerous drawbacks to this approach: the determination and effect of the weights in the cost function are difficult to ascertain, imperative requirements are difficult to incorporate and make the optimization process more complex, and the definition of the performance indices is not trivial, to name but a few. The main issues can be stated as:

- the robustness of the design solution obtained with the cost function approach with respect to the uncertainties in the final design. Indeed the real instantiation of a theoretical solution will always differ from the latter because of the manufacturing tolerances and other uncertainties that are inherent to a mechanical system,
- performance requirements may be antagonistic (e.g., workspace and accuracy) and the optimal design approach only provides a compromise between these requirements that is difficult to master through the weights.

An alternative to optimal design is referred to as *appropriate design*, in which no optimization is considered but the objective is to ensure that desired requirements are satisfied. This approach is based on the definition of the *parameter space* in which each dimension is associated with a design parameter. Performance requirements are considered in turn and the regions of the parameter space that correspond to robots satisfying the requirements are calculated. The design solution is then obtained as the intersection of the regions obtained for each individual requirement.

In practice, only an approximation of the regions is necessary since values close to the boundary cannot be physically realized due to manufacturing tolerances. For that calculation, interval analysis was successfully used in various applications [12.20, 118].

The appropriate design approach is clearly more complex to implement than the cost function approach but has the advantage of providing all design solutions, with the consequence that manufacturing tolerances may be taken into account to ensure that the real robot will also satisfy the desired requirements.

12.10 Application Examples

Parallel robots have been successfully used in many applications, a few of which will now be briefly mentioned. Almost all recent land-based telescopes use parallel robots, either as a secondary mirror alignment system (for example the University of Arizona MMT or the European Organisation for Astronomical Research in the Southern Hemisphere (ESO) Visible and Infrared Survey Telescope for Astronomy (VISTA)) or as a primary mirror pointing device. A wire-driven parallel robot flew in the space shuttle mission STS-63 in February 1999 while an octopod (a parallel robot with eight legs) was used to isolate the space shuttle payload from vibration. A parallel platform is used in the medical rehabilitation platform Caren of Motek. All flight simulators use a parallel structure as a motion platform. They are nowadays used for driving simulators as well [12.119].

In industry, numerous machine tools based on parallel structures have been designed: some of them have found a niche market (e.g., the Tricept) and it can be expected that more will be used in the future (especially for high-speed manufacturing) as soon as a controller specifically designed for parallel structures is developed (current controllers are basically identical to the ones used for the classical linear machine tools and hence do not allow the potential of parallel structures to be fully exploited). Ultra-accurate positioning devices based on parallel robots are proposed by companies such as Physik Instrumente, Micos, and Alio. In the food industry the Delta robot proposed by Demaurex and SIG Robotics is widely used for fast packaging. Other companies such as Adept are also developing fast parallel robots with three and four degrees of freedom.

12.11 Conclusion and Further Reading

The analysis of a parallel mechanism may be partly based on methods that are presented in other chapters of this handbook:

- *kinematics*: kinematics background is covered in Chap. 1,
- *dynamics*: general approaches are presented in Chap. 2 while identification of the dynamic parameters is addressed in Chap. 14,
- *design*: design methodologies are covered in Chap. 10,
- *control*: control issues are presented in Chaps. 5, 6, and 7 although the closed-loop structure of parallel robots may require some adaptation of the control schemes.

It must also be emphasized that efficient numerical analysis is a key point for many algorithms related to parallel robots. System solving with Gröbner basis, continuation method, and interval analysis is essential for kinematics, workspace, and singularity analysis.

Further information and up-to-date extensive references and papers related to parallel robots can be found on the following two web sites [12.120, 121]

Useful complementary readings on parallel robots are [12.9, 122, 123].

Parallel robots are slowly finding their way into various applications, not only in industry but also in field and service robotics, as discussed in Part F of this handbook. Still, compared to their serial counterpart, their analysis is far from being complete.

References

12.1 J.E. Gwinnett: Amusement device, Patent 1789680 (1931)

12.2 V.E. Gough: Contribution to discussion of papers on research in automobile stability, control and tyre performance, 1956–1957. Proc. Auto Div. Inst. Mech. Eng

12.3 D. Stewart: A platform with 6 degrees of freedom, Proc. Inst. Mech. Eng. **180**(1,15), 371–386 (1965)

12.4 I.A. Bonev: The true origins of parallel robots. http://www.parallemic.org/Reviews /Review007.html , (2003)

12.5 J. Albus, R. Bostelman, N. Dagalakis: The NIST ROBOCRANE, J. Robot. Syst. **10**(5), 709–724 (1993)

12.6 R. Clavel: DELTA, a fast robot with parallel geometry. In: *18th Int. Symp. on Industrial Robots (ISIR)*, Lausanne, (1988) pp. 91–100

12.7 K.H. Hunt: Structural kinematics of in parallel actuated robot arms, J. Mech. Transmiss. Automation Design **105**(4), 705–712 (1983)

12.8 J.M. Hervé: Group mathematics and parallel link mechanisms. In: *9th IFToMM World Congress on the Theory of Machines and Mechanisms*, Milan, (1995) pp. 2079–2082

12.9 X. Kong, C.M. Gosselin: *Type synthesis of parallel mechanisms* (Springer Tracts in Advanced Robotics, Heidelberg 2007), Vol. 33

12.10 www-sop.inria.fr/coprin/equipe/merlet/Archi /archirobot.html

12.11 P. Dietmaier: The Stewart-Gough platform of general geometry can have 40 real postures. In: *ARK*, Strobl, (1998) pp. 7–16

12.12 M.L. Husty: An algorithm for solving the direct kinematic of Stewart-Gough-type platforms, Mechanism Machine Theory **31**(4), 365–380 (1996)

12.13 M. Raghavan: The Stewart platform of general geometry has 40 configurations, ASME J. Mech. Des. **115**(2), 277–282 (1993)

12.14 J.C. Faugère, D. Lazard: The combinatorial classes of parallel manipulators, Mechanism Machine Theory **30**(6), 765–776 (1995)

12.15 T.-Y. Lee, J.-K. Shim: Improved dyalitic elimination algorithm for the forward kinematics of the general Stewart-Gough platform, Mechanism Machine Theory **38**(6), 563–577 (2003)

12.16 F. Rouillier: Real roots counting for some robotics problems. In: *Computational Kinematics*, ed. by J.-P. Merlet, B. Ravani (Kluwer, Dordrecht 1995) pp. 73–82

12.17 J.-P. Merlet: Solving the forward kinematics of a Gough-type parallel manipulator with interval analysis, Int. J. Robot. Res. **23**(3), 221–236 (2004)

12.18 D. Manocha: Algebraic and Numeric Techniques for Modeling and Robotics. Ph.D. Thesis (University of California, Berkeley 1992)

12.19 A.J. Sommese, C.W. Wampler: *Numerical Solutions of Polynomial Systems Arising in Engineering and Science* (World Scientific, Singapore 2005)

12.20 J-P. Merlet, D. Daney: Dimensional synthesis of parallel robots with a guaranteed given accuracy over a specific workspace. In: *IEEE Int. Conf. on Robotics and Automation*, Barcelona, (2005)

12.21 L. Baron, J. Angeles: The direct kinematics of parallel manipulators under joint-sensor redundancy, IEEE Trans. Robot. Automat. **16**(1), 12–19 (2000)

12.22 J-P. Merlet: Closed-form resolution of the direct kinematics of parallel manipulators using extra sensors data. In: *IEEE Int. Conf. on Robotics and Automation*, Atlanta, (1993) pp. 200–204

12.23 R. Stoughton, T. Arai: Kinematic optimization of a chopsticks-type micro-manipulator. In: *Japan-USA Symp. on Flexible Automation*, San Fransisco, (1993) pp. 151–157

12.24 J.-P. Merlet: Jacobian, manipulability,condition number, and accuracy of parallel robots. ASME J. Mech. Design **128**(1), 199–206 (2006)

12.25 F-T. Niaritsiry, N. Fazenda, R. Clavel: Study of the source of inaccuracy of a 3 dof flexure hinge-based parallel manipulator. In: *IEEE Int. Conf. on Robotics and Automation*, New Orleans, (2004) pp. 4091–4096

12.26 G. Pritschow, C. Eppler, T. Garber: Influence of the dynamic stiffness on the accuracy of PKM. In: *3rd Chemnitzer Parallelkinematik Seminar*, Chemnitz, (2002) pp. 313–333

12.27 V. Parenti-Castelli, S. Venanzi: On the joint clearance effects in serial and parallel manipulators. In: *Workshop on Fundamental Issues and Future Research Directions for Parallel Mechanisms and Manipulators*, Québec, (2002) pp. 215–223

12.28 A. Pott, M. Hiller: A new approach to error analysis in parallel kinematic structures. In: *ARK*, Sestri-Levante, (2004)

12.29 K. Wohlhart: Degrees of shakiness, Mechanism Machine Theory **34**(7), 1103–1126 (1999)

12.30 K. Tönshoff, B. Denkena, G. Günther, H.-C. Möhring: Modelling of error effects on the new hybrid kinematic DUMBO structure. In: *3rd Chemnitzer Parallelkinematik Seminar*, Chemnitz, (2002) pp. 639–653

12.31 U. Sellgren: Modeling of mechanical interfaces in a systems context. In: *Int. ANSYS Conf.*, Pittsburgh, (2002)

12.32 W. Khalil, S. Besnard: Identificable parameters for the geometric calibration of parallel robots, Arch. Control Sci. **11**(3-4), 263–277 (2001)

12.33 C.W. Wampler, J.M. Hollerbach, T. Arai: An implicit loop method for kinematic calibration and its application to closed-chain mechanisms, IEEE Trans. Robot. Autom. **11**(5), 710–724 (1995)

12.34 D. Daney, Y. Papegay, A. Neumaier: Interval methods for certification of the kinematic calibration of parallel robots. In: *IEEE Int. Conf. on Robotics and Automation*, New Orleans, (2004) pp. 1913–1918

12.35 D. Daney, N. Andreff, Y. Papegay: Interval method for calibration of parallel robots: a vision-based experimentation. In: *Computational Kinematics*, Cassino, (2005)

12.36 A. Nahvi, J.M. Hollerbach: The noise amplification index for optimal pose selection in robot calibration. In: *IEEE Int. Conf. on Robotics and Automation*, Minneapolis, (1996) pp. 647–654

12.37 G. Meng, L. Tiemin, Y. Wensheng: Calibration method and experiment of Stewart platform using a laser tracker. In: *Int. Conf on Systems, Man and Cybernetics*, The Hague, (2003) pp. 2797–2802

12.38 D. Daney: Optimal measurement configurations for Gough platform calibration. In: *IEEE Int. Conf. on Robotics and Automation*, Washington, (2002) pp. 147–152

12.39 C. Gosselin, J. Angeles: Singularity analysis of closed-loop kinematic chains, IEEE Trans. Robot. Automation **6**(3), 281–290 (1990)

12.40 D. Zlatanov, R.G. Fenton, B. Benhabib: A unifying framework for classification and interpretation of mechanism singularities, ASME J. Mech. Des. **117**(4), 566–572 (1995)

12.41 D. Zlatanov, I.A. Bonev, C.M. Gosselin: Constraint singularities of parallel mechanisms. In: *IEEE Int. Conf. on Robotics and Automation*, Washington, (2002) pp. 496–502

12.42 G. Liu, Y. Lou, Z. Li: Singularities of parallel manipulators: a geometric treatment, IEEE Trans. Robot. Autom. **19**(4), 579–594 (2003)

12.43 I.A. Bonev, D. Zlatanov: The mystery of the singular SNU translational parallel robot. www.parallemic.org/Reviews/Review004.html, (2001)

12.44 H. Li., C.M. Gosselin, M.J. Richard, B. St-Onge Mayer: Analytic form of the six-dimensional singularity locus of the general Gough-Stewart platform, ASME J. Mech. Des. **128**(1), 279–287 (2006)

12.45 B. St-Onge Mayer, C.M. Gosselin: Singularity analysis and representation of the general Gough-Stewart platform, Int. J. Robot. Res. **19**(3), 271–288 (2000)

12.46 J.-P. Merlet: Singular configurations of parallel manipulators and Grassmann geometry, Int. J. Robot. Res. **8**(5), 45–56 (1989)

12.47 J-P. Merlet: On the infinitesimal motion of a parallel manipulator in singular configurations. In: *IEEE Int. Conf. on Robotics and Automation*, Nice, (1992) pp. 320–325

12.48 H. Pottmann, M. Peternell, B. Ravani: Approximation in line space. Applications in robot kinematics. In: *ARK*, Strobl, (1998) pp. 403–412

12.49 P.A. Voglewede, I. Ebert-Uphoff: Measuring "closeness" to singularities for parallel manipulators. In: *IEEE Int. Conf. on Robotics and Automation*, New Orleans, (2004) pp. 4539–4544

12.50 J.-P. Merlet, D. Daney: A formal-numerical approach to determine the presence of singularity within the workspace of a parallel robot. In: *Computational Kinematics*, ed. by F.C. Park, C.C. Iurascu (EJCK, Seoul 2001) pp. 167–176

12.51 S. Bhattacharya, H. Hatwal, A. Ghosh: Comparison of an exact and an approximate method of singularity avoidance in platform type parallel manipulators, Mechanism Machine Theory **33**(7), 965–974 (1998)

12.52 D.N. Nenchev, M. Uchiyama: Singularity-consistent path planning and control of parallel robot motion through instantaneous-self-motion type. In: *IEEE*

Int. Conf. on Robotics and Automation, Minneapolis, (1996) pp. 1864–1870

12.53 R. Ranganath, P.S. Nair, T.S. Mruthyunjaya, A. Ghosal: A force–torque sensor based on a Stewart platform in a near-singular configuration, Mechanism Machine Theory **39**(9), 971–998 (2004)

12.54 M.L. Husty, A. Karger: Architecture singular parallel manipulators and their self-motions. In: *ARK*, Piran, (2000) pp. 355–364

12.55 A. Karger: Architecture singular planar parallel manipulators, Mechanism Machine Theory **38**(11), 1149–1164 (2003)

12.56 K. Wohlhart: Mobile 6-SPS parallel manipulators, J. Robot. Syst. **20**(8), 509–516 (2003)

12.57 C. Innocenti, V. Parenti-Castelli: Direct kinematics of the 6-4 fully parallel manipulator with position and orientation uncoupled. In: *European Robotics and Intelligent Systems Conf.*, Corfou, (1991)

12.58 G. Gogu: Mobility of mechanisms: a critical review, Mechanism Machine Theory **40**(10), 1068–1097 (2005)

12.59 I. Zabalza, J. Ros, J.J. Gil, J.M. Pintor, J.M. Jimemenz: Tri-Scott. A new kinematic structure for a 6-dof decoupled parallel manipulator. In: *Workshop on Fundamental Issues and Future Research Directions for Parallel Mechanisms and Manipulators*, Québec, (2002) pp. 12–15

12.60 C. Gosselin: Determination of the workspace of 6-dof parallel manipulators, ASME J. Mech. Des. **112**(3), 331–336 (1990)

12.61 J-P. Merlet: Geometrical determination of the workspace of a constrained parallel manipulator. In: *ARK*, Ferrare, (1992) pp. 326–329

12.62 F.A. Adkins, E.J. Haug: Operational envelope of a spatial Stewart platform, ASME J. Mech. Des. **119**(2), 330–332 (1997)

12.63 E.J. Haugh, F.A. Adkins, C.M. Luh: Operational envelopes for working bodies of mechanisms and manipulators, ASME J. Mech. Des. **120**(1), 84–91 (1998)

12.64 J.-P. Merlet: Determination of 6D workspaces of Gough-type parallel manipulator and comparison between different geometries, Int. J. Robot. Res. **18**(9), 902–916 (1999)

12.65 P. Wenger, D. Chablat: Workspace and assembly modes in fully parallel manipulators: a descriptive study. In: *ARK*, Strobl, (1998) pp. 117–126

12.66 J. Hesselbach, C. Bier, A. Campos, H. Löwe: Parallel robot specific control fonctionalities. In: *2nd Int. Colloquium, Collaborative Research Centre 562*, Braunschweig, (2005) pp. 93–108

12.67 J-P. Merlet: An efficient trajectory verifier for motion planning of parallel machine. In: *Parallel Kinematic Machines Int. Conf.*, Ann Arbor, (2000)

12.68 J. Cortés, T. Siméon: Probabilistic motion planning for parallel mechanisms. In: *IEEE Int. Conf. on Robotics and Automation*, Taipei, (2003) pp. 4354–4359

12.69 J.H. Yakey, S.M. LaValle, L.E. Kavraki: Randomized path planning for linkages with closed kinematic chains, IEEE Trans. Robot. Autom. **17**(6), 951–958 (2001)

12.70 J-P. Merlet, M-W. Perng, D. Daney: Optimal trajectory planning of a 5-axis machine tool based on a 6-axis parallel manipulator. In: *ARK*, Piran, (2000) pp. 315–322

12.71 D. Shaw, Chen Y-S. Cutting path generation of the Stewart platform-based milling machine using an end-mill, Int. J. Prod. Res. **39**(7), 1367–1383 (2001)

12.72 Z. Wang, Z. Wang, W. Liu, Y. Lei: A study on workspace, boundary workspace analysis and workpiece positioning for parallel machine tools, Mechanism Machine Theory **36**(6), 605–622 (2001)

12.73 D.R. Kerr: Analysis, properties, and design of a Stewart-platform transducer, J. Mech. Transmiss. Autom. Des. **111**(1), 25–28 (1989)

12.74 C.C. Nguyen, S.S. Antrazi, Z.L. Zhou: Analysis and experimentation of a Stewart platform-based force/torque sensor, Int. J. Robot. Autom. **7**(3), 133–141 (1992)

12.75 C. Reboulet, A. Robert: Hybrid control of a manipulator with an active compliant wrist. In: *3rd ISRR*, Gouvieux, France, (1985) pp. 76–80

12.76 J. Duffy: *Statics and Kinematics with Applications to Robotics* (Cambridge University Press, New-York 1996)

12.77 C. Huang, W-H. Hung, I. Kao: New conservative stiffness mapping for the Stewart-Gough platform. In: *IEEE Int. Conf. on Robotics and Automation*, Washington, (2002) pp. 823–828

12.78 J.L. Herder: *Energy-Free Systems: Theory, Conception and Design of Statically Balanced Spring Mechanisms*. Ph.D. Thesis, Delft University of Technology, Delft, (2001)

12.79 G.R. Dunlop, T.P. Jones: Gravity counter balancing of a parallel robot for antenna aiming. In: *6th ISRAM*, Montpellier, (1996) pp. 153–158

12.80 M. Jean, C. Gosselin: Static balancing of planar parallel manipulators. In: *IEEE Int. Conf. on Robotics and Automation*, Minneapolis, (1996) pp. 3732–3737

12.81 I. Ebert-Uphoff, C.M. Gosselin, T. Laliberté: Static balancing of spatial parallel platform-revisited, ASME J. Mech. Des. **122**(1), 43–51 (2000)

12.82 C.M. Gosselin, J. Wang: Static balancing of spatial six-degree-of-freedom parallel mechanisms with revolute actuators, J. Robot. Syst. **17**(3), 159–170 (2000)

12.83 M. Leblond, C.M. Gosselin: Static balancing of spatial and planar parallel manipulators with prismatic actuators. In: *ASME Design Engineering Technical Conferences*, Atlanta, (1998)

12.84 B. Monsarrat, C.M. Gosselin: Workspace analysis and optimal design of a 3-leg 6-DOF parallel platform mechanism, IEEE Trans. Robot. Autom. **19**(6), 954–966 (2003)

12.85 J. Wang, C.M. Gosselin: Static balancing of spatial three-degree-of-freedom parallel mechanisms, Mechanism Machine Theory **34**(3), 437–452 (1999)
12.86 Y. Wu, C.M. Gosselin: Synthesis of reactionless spatial 3-dof and 6-dof mechanisms without separate counter-rotations, Int. J. Robot. Res. **23**(6), 625–642 (2004)
12.87 M. Ait-Ahmed: Contribution à la modélisation géométrique et dynamique des robots parallèles. Ph.D. Thesis (Université Paul Sabatier, Toulouse 1993)
12.88 M.-J. Liu, C.-X. Li, C.-N. Li: Dynamics analysis of the Gough-Stewart platform manipulator, IEEE Trans. Robot. Autom. **16**(1), 94–98 (2000)
12.89 G.F. Liu, X.Z. Wu, Z.X. Li: Inertial equivalence principle and adaptive control of redundant parallel manipulators. In: *IEEE Int. Conf. on Robotics and Automation*, Washington, (2002) pp. 835–840
12.90 R. Clavel: Conception d'un robot parallèle rapide à 4 degrés de liberté. Ph.D. Thesis (EPFL, Lausanne 1991), n° 925
12.91 J. Gallardo, J.M. Rico, A. Frisoli, D. Checcacci, M. Bergamasco: Dynamics of parallel manipulators by means of screw theory, Mechanism Machine Theory **38**(11), 1113–1131 (2003)
12.92 L.-W. Tsai: Solving the inverse dynamics of a Stewart-Gough manipulator by the principle of virtual work, ASME J. Mech. Des. **122**(1), 3–9 (2000)
12.93 J. Wang, C.M. Gosselin: A new approach for the dynamic analysis of parallel manipulators, Multibody Syst. Dyn. **2**(3), 317–334 (1998)
12.94 Z. Geng, L.S. Haynes: On the dynamic model and kinematic analysis of a class of Stewart platforms, Robot. Auton. Syst. **9**(4), 237–254 (1992)
12.95 K. Liu, F. Lewis, G. Lebret, D. Taylor: The singularities and dynamics of a Stewart platform manipulator, J. Intell. Robot. Syst. **8**, 287–308 (1993)
12.96 K. Miller, R. Clavel: The Lagrange-based model of Delta-4 robot dynamics, Robotersysteme **8**(1), 49–54 (1992)
12.97 K. Miller: Optimal design and modeling of spatial parallel manipulators, Int. J. Robot. Res. **23**(2), 127–140 (2004)
12.98 A. Codourey, E. Burdet A body oriented method for finding a linear form of the dynamic equations of fully parallel robot. In: *IEEE Int. Conf. on Robotics and Automation*, Albuquerque, (1997) pp. 1612–1618
12.99 B. Dasgupta, P. Choudhury: A general strategy based on the Newton–Euler approach for the dynamic formulation of parallel manipulators, Mechanism Machine Theory **34**(6), 801–824 (1999)
12.100 P. Guglielmetti: Model-Based control of fast parallel robots: a global approach in operational space. Ph.D. Thesis (EPFL, Lausanne 1994)
12.101 K. Harib, K. Srinivasan: Kinematic and dynamic analysis of Stewart platform-based machine tool structures, Robotica **21**(5), 541–554 (2003)
12.102 W. Khalil, O. Ibrahim: General solution for the dynamic modeling of parallel robots. In: *IEEE Int. Conf. on Robotics and Automation*, New Orleans, (2004) pp. 3665–3670
12.103 C. Reboulet, T. Berthomieu: Dynamic model of a six degree of freedom parallel manipulator. In: *ICAR*, Pise, (1991) pp. 1153–1157
12.104 H. Abdellatif, B. Heimann: Adapted time-optimal trajectory planning for parallel with full dynamic modelling. In: *IEEE Int. Conf. on Robotics and Automation*, Barcelona, (2005) pp. 413–418
12.105 S. Tadokoro: Control of parallel mechanisms, Adv. Robot. **8**(6), 559–571 (1994)
12.106 M. Honegger, A. Codourey, E. Burdet: Adaptive control of the Hexaglide, a 6 dof parallel manipulator. In: *IEEE Int. Conf. on Robotics and Automation*, Albuquerque, (1997) pp. 543–548
12.107 S. Bhattacharya, H. Hatwal, A. Ghosh: An on-line estimation scheme for generalized Stewart platform type parallel manipulators, Mechanism Machine Theory **32**(1), 79–89 (1997)
12.108 J. Hesselbach, O. Becker, M. Krefft, I. Pietsch, N. Plitea: Dynamic modelling of plane parallel robot for control purposes. In: *3rd Chemnitzer Parallelkinematik Seminar*, Chemnitz, (2002) pp. 391–409
12.109 P. Guglielmetti, R. Longchamp: A closed-form inverse dynamics model of the Delta parallel robot. In: *4th IFAC Symp. on Robot Control*, *Syroco*, Capri, (1994) pp. 51–56
12.110 K. Miller: Modeling of dynamics and model-based control of DELTA direct-drive parallel robot, J. Robot. Mechatron. **17**(4), 344–352 (1995)
12.111 E. Burdet, M. Honegger, A. Codourey: Controllers with desired dynamic compensation and their implementation on a 6 dof parallel manipulator. In: *IEEE Int. Conf. on Intelligent Robots and Systems (IROS)*, Takamatsu, (2000)
12.112 K. Yamane, M. Okada, N. Komine, Y. Nakamura: Parallel dynamics computation and h_∞ acceleration control of parallel manipulators for acceleration display, J. Dyn. Syst. Meas. Control **127**(2), 185–191 (2005)
12.113 J.E. McInroy: Modeling and design of flexure jointed Stewart platforms for control purposes, IEEE/ASME Trans. Mechatron. **7**(1), 95–99 (2002)
12.114 F. Xi: Dynamic balancing of hexapods for high-speed applications, Robotica **17**(3), 335–342 (1999)
12.115 J. Angeles The robust design of parallel manipulators. In: *1st Int. Colloquium*, *Collaborative Research Centre 562*, Braunschweig, (2002) pp. 9–30
12.116 S. Bhattacharya, H. Hatwal, A. Ghosh: On the optimum design of a Stewart platform type parallel manipulators, Robotica **13**(2), 133–140 (1995)

12.117 K.E. Zanganeh, J. Angeles: Kinematic isotropy and the optimum design of parallel manipulators, Int. J. Robot. Res. **16**(2), 185–197 (1997)
12.118 H. Fang, J.-P. Merlet: Multi-criteria optimal design of parallel manipulators based on interval analysis, Mechanism Machine Theory **40**(2), 151–171 (2005)
12.119 www-nads-sc.uiowa.edu
12.120 www.sop.inria.fr/coprin/equipe/merlet/merlet_eng.html
12.121 www.parallemic.org
12.122 J.-P. Merlet: *Parallel Robots* (Springer, Heidelberg 2006)
12.123 L.-W. Tsai: *Robot Analysis: The Mechanics of Serial and Parallel Manipulators* (Wiley, Hoboken 1999)