Chapter 1

Introduction: Overview of parallel manipulators and literature review

1.1 Introduction

A parallel manipulator can be defined as

> a closed-loop kinematic chain mechanism whose end-effector is linked to the base by several independent kinematic chains (Merlet [1]).

Parallel manipulators have been increasingly studied and developed over the last couple of decades (Merlet [2], Dasgupta and Mruthyunjaya [3]) from both a theoretical viewpoint as well as for practical applications. Parallel structures are certainly not a new discovery, however advances in computer technology and development of sophisticated control techniques, amongst other factors, have allowed for the more recent practical implementation of

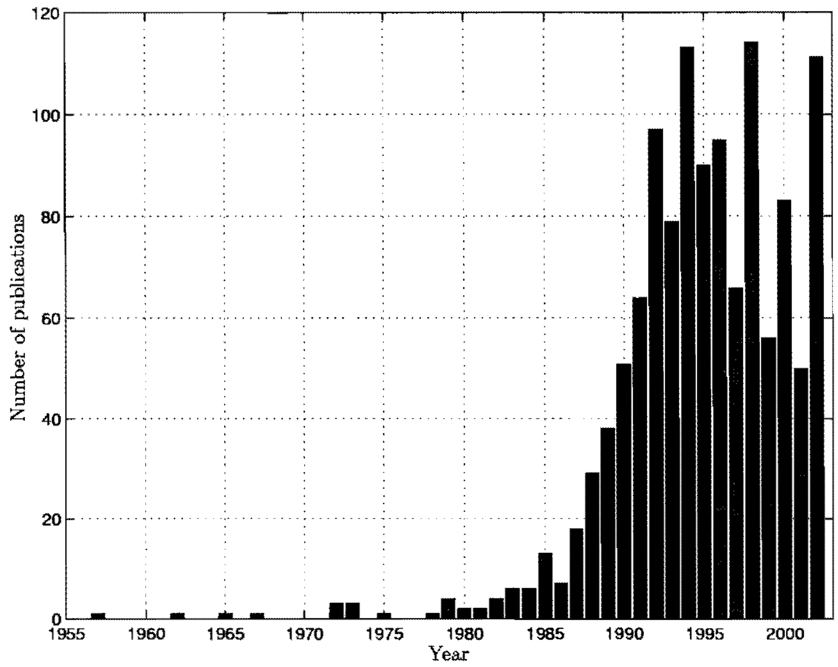

Figure 1.1: Parallel manipulator publications by year 1955-2002

parallel manipulators. This trend is well illustrated by the ever increasing number of publications dedicated to parallel manipulators. Figure 1.1 shows the approximate annual numbers of publications related to parallel manipulators for the past 50 years as reported by Merlet [4].

Interest in parallel manipulators has been stimulated by the advantages offered over traditional serial manipulator architectures. In fact, there exists an interesting duality between parallel and serial architectures, both in terms of analysis and performance, where parallel manipulators have good characteristics in areas where serial manipulators perform poorly, and vice versa. Zamanov and Sotirov [5], Waldron and Hunt [6] and Duffy [7] seek to explain this duality and Fichter and MacDowell [8] discuss some of the practical issues relating to performance of serial and parallel robots.

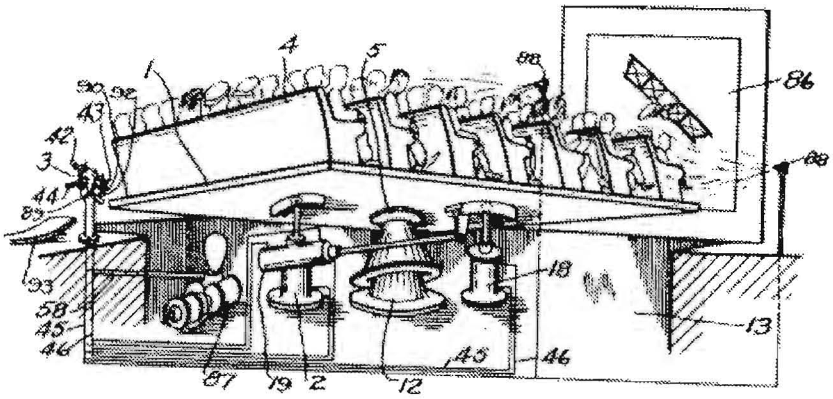

Figure 1.2: The spherical parallel mechanism patented by Gwinnett

The literature review presented in this chapter provides an overview of the development of parallel manipulators, workspace determination, optimal design, characterization of manipulator performance, and optimization methods. This selective overview is used in Section 1.6 to motivate the current study.

1.2 Brief history of parallel manipulator development

Some theoretical problems associated with parallel structures were mentioned by the English architect Sir Christopher Wren as early as the 17^{th} century. Cauchy, Lebesgue, Bricard and Borel all published papers on problems related to parallel mechanisms in the 19^{th} and early 20^{th} century (Merlet [1]).

It appears that the first practical application for a parallel manipulator was proposed by Gwinnett [9] who was granted a patent in 1931 for a motion platform, based on a spherical parallel mechanism (Bonev [10]). As illustrated in Figure 1.2, the motion platform was intended for use in the entertainment industry.

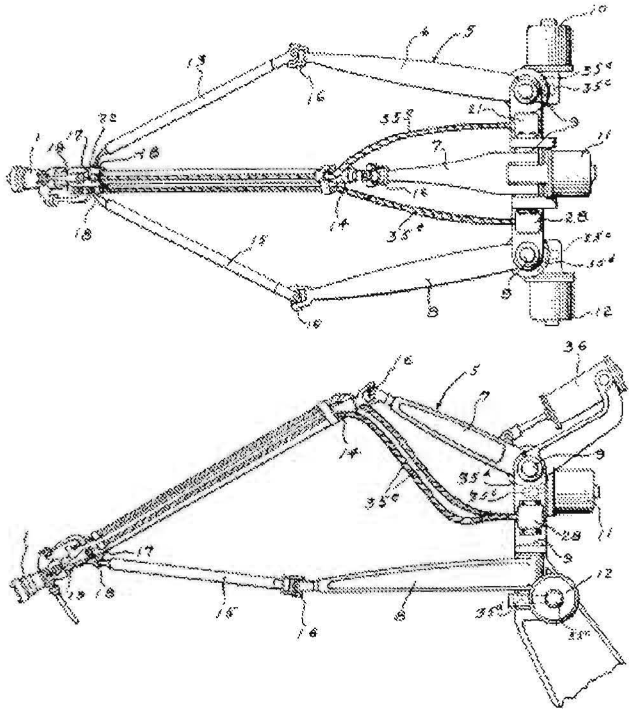

Figure 1.3: Pollard's spatial industrial robot

In 1942 a patent was issued to Pollard [11] for what is now known as the first industrial parallel robot design. The robot, shown in Figure 1.3, was intended for spray painting, but was never built.

In 1965 Stewart [12], published a paper in which he proposed a six-degree-of-freedom (six-dof) parallel platform for use as a flight simulator. This paper attracted so much attention that many researchers began referring to octahedral hexapod parallel manipulators as "Stewart platforms". It is somewhat ironic though that similar ideas to Stewart's had already been independently conceived by two other researchers.

Eric Gough, an employee of the Dunlop Rubber Co., England, had constructed an octahedral hexapod in 1954 (Gough [13], see communications from [12]). This parallel manipulator was used as a universal tyre testing machine as shown in Figure 1.4. Interestingly, Bonev [10] notes that this

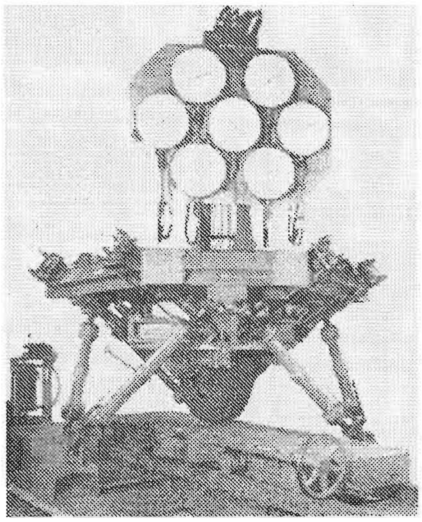

Figure 1.4: The universal tyre-testing machine of Gough

machine continued to operate until 1999.

At the same time an American engineer, Klaus Cappel, was also independently developing an octahedral hexapod manipulator. A patent for an octahedral hexapod to be used as a motion simulator was filed in 1964 and issued in 1967 [14]. The first ever flight simulator based on a octahedral hexapod was made under licence from this patent. Figure 1.5 shows a drawing taken from Cappel's patent.

Since these early days, parallel manipulators have proliferated and found application in many fields. One of the most promising applications is in the manufacturing industry. Prominent early examples of machine tools based on parallel architectures are the Giddings and Lewis Variax and the Ingersoll Octahedral Hexapod which were both first presented at the 1994 International Manufacturing Technology Show (IMTS) in Chicago. More recent and successful applications of parallel manipulator architectures have been the Z^3 machining head developed by DS Technologie Gmbh (DST) which is shown in Figure 1.6 and the Neos Tricept. Other applications include flight simulators, fine positioning devices, overhead cranes (when using cabledriven manipulator architectures) and more recently in medical applications as surgical robots.

1.3 Workspace determination of parallel manipulators

The *workspace* of a manipulator may loosely be defined as

> regions [in output space[1]] which can be reached by a reference point located on the mobile platform [of the manipulator] (Merlet *et al.* [15]).

Based on this definition, the workspace of any manipulator has the same dimension as the number of output degrees of freedom of the manipulator. For example, the workspace of a planar parallel manipulator manipulator, which has three degrees of freedom (translations x and y in the plane and rotation ϕ about the out of plane axis) is thus most fully represented three-dimensionally, with two axes used to represent the x and y positional coordinates of the reference point, and the third axis corresponding to the angular orientation ϕ of the moving platform.

Similarly, the workspace of the Cappell, Gough and Stewart platforms (see the previous section) can only really be fully described in six-dimensional space. Of course, it is difficult for us to conceptualize any space of dimension greater than three. In order to obtain descriptions of manipulator positioning capabilities that can be easily visualized and understood, subsets of the full workspace are defined for which restrictions are placed on some of the output degrees of freedom of the manipulator, most commonly on the orientation

[1]output space – positional plus angular orientational dimensions

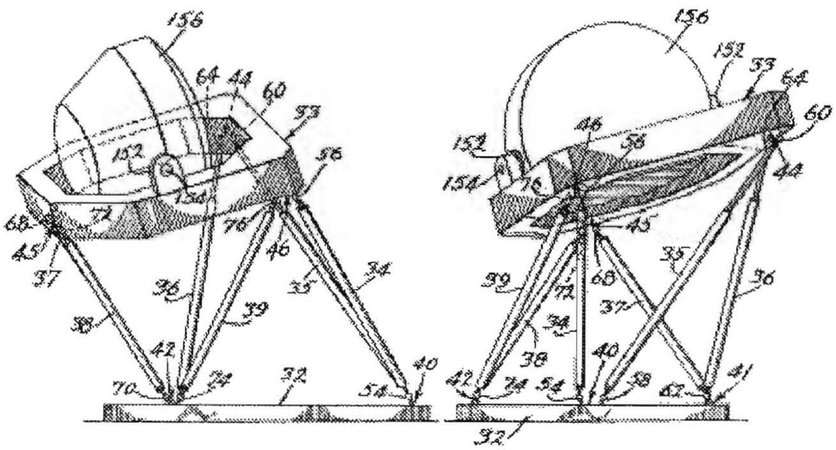

Figure 1.5: Octahedral hexapod motion simulator by Cappel

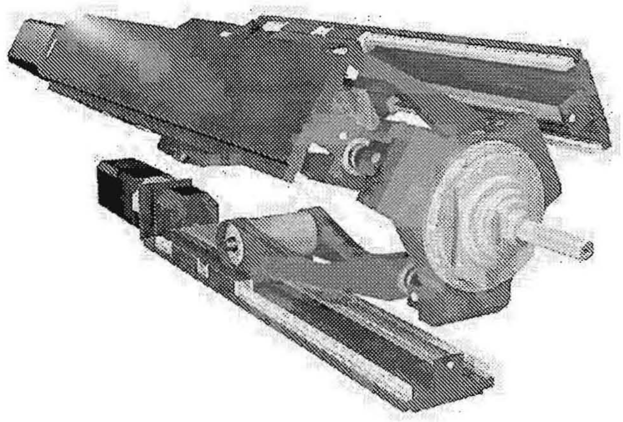

Figure 1.6: The Z^3 machining head by DST

of the moving platform. The various types of workspace commonly used are defined in Section 1.3.1.

Determination of these workspaces for parallel manipulators poses a challenging problem (Merlet *et al.* [15]). Various methods proposed by different researchers are discussed in Section 1.3.2.

1.3.1 Classification of workspace types

Constant orientation workspace

A specific *constant [angular] orientation workspace* is defined as (Merlet *et al.* [15])

> the positional region which can be reached by the reference point of the manipulator when the mobile platform has a specific prescribed constant [angular] orientation.

The constant orientation[2] workspace has the same dimension as the number of translational [positional] output degrees of freedom of the parallel manipulator.

For the planar manipulators studied here, the constant orientation workspace is two-dimensional, and will be denoted as $W^C[\phi^{\text{fix}}]$, where ϕ^{fix} is the specific prescribed and fixed angular orientation of the moving platform associated with that particular constant orientation workspace.

Maximal workspace

The definition of the *maximal workspace* adopted here is

[2]Hereafter 'orientation' will be considered synonymous with 'angular orientation'.

> the positional region which can be reached by the reference point of the manipulator with no restrictions on the orientation of the moving platform.

This is in agreement with the definition offered by Merlet *et al.* [15]. Haug *et al.* [16] also refer to maximal workspaces as "accessible output sets". Essentially, the maximal workspace can be thought of as either the projection of the full output workspace onto the positional space of the manipulator, or the *union* of all possible constant orientation workspaces.

For planar manipulators, the maximal workspace, denoted W^M, is two dimensional and for spatial manipulators, three-dimensional.

Dextrous workspace

The *dextrous workspace* of a manipulator is

> the region reachable by the reference point of the manipulator with all orientations in a given set $[\boldsymbol{\phi}^{\min}, \boldsymbol{\phi}^{\max}]$.

This terminology in consistent with that used by Haug *et al.* [17] and Du Plessis and Snyman [18], but differs slightly from that used by Merlet *et al.* [15] who use the term "total orientation workspace" to describe this workspace, and the term "dextrous workspace" for the special case where the moving platform is required to reach *all* possible orientations.

Once again the dextrous workspace is two-dimensional for the planar case and three-dimensional for the spatial case. The dextrous workspace is denoted $W^D[\phi^{\min}, \phi^{\max}]$ here for the planar case, and can also be thought of as the *intersection* of all constant orientation workspaces in the *orientation interval* $[\phi^{\min}, \phi^{\max}]$.

Orientation workspace

Finally a specific *orientation workspace* is defined as

> the set of angular orientations (orientational region) which can be attained by the moving platform for a fixed position of the reference point.

The orientation workspace is difficult to represent for a general spatial manipulator. Merlet [19] notes that simply plotting the standard Euler angles does not lead to intelligible results. He suggests mapping instead the positions which can be reached by a unit vector, fixed to the moving platform, onto a unit sphere. Bonev *et al.* [20] suggest the use of modified Euler tilt-and-torsion angles, which result in a compact and intuitive representation of the orientation workspace. For the planar manipulator, the orientation workspace is one-dimensional and is denoted $W^O[\mathbf{u}^{\text{fix}}]$, where $\mathbf{u}^{\text{fix}}$ is a vector containing the fixed position of the reference point.

1.3.2 Methods for workspace determination

In general, determination of parallel manipulator workspaces poses a more challenging problem than for serial manipulators. This is because of the strong coupling of the positional and orientational capabilities of parallel manipulators. Merlet [1] gives the example of a six-dof serial robot with a concurrent axis wrist. For this manipulator the three-dimensional volume, which the robot can reach, depends only on the motion capability of the first three actuated joints, while the orientational ability depends only on the last three joints. Compare this to a hexapod, where orientational and positional ability are influenced simultaneously by all the actuators. The most prominent and commonly-used methods for workspace determination can be grouped into four categories.

Geometrical methods

Geometrical methods are based on the observation that the workspace boundary must necessarily always be associated with a physical limit on the manipulator input kinematic chains. By separately taking into account the constraints on each input kinematic chain and geometrically determining the region which can be reached by reference point under these conditions, and then determining the intersection of all these regions, the actual workspace of the manipulator can be determined.

The method as applied to parallel manipulators was first introduced by Gosselin and Angeles [21] who used a geometrical approach to determine constant orientation workspaces of a planar 3-$\underline{R}RR$ parallel manipulator. In this notation, the number signifies the number of kinematic chains linking the moving platform to the base, and the set of letters defines the sequence of joints used in each kinematic chain. A revolute joint is denoted by R and a prismatic joint by P. Spatial universal and spherical joints are respectively denoted by U (or sometimes RR) and S. Actuated joints are indicated by underlining.

The geometrical methodology, including the effects of passive joint limits, is applied to 3-$R\underline{P}R$ planar manipulators in Gosselin and Jean [22]. Merlet *et al.* [15] extend the methodology to determining other types of workspaces of planar 3-$R\underline{P}R$ parallel manipulators (see Section 1.3.1).

Geometric methods have also been used to determine constant orientation workspaces of more complex 6-$U\underline{P}S$ spatial manipulators by Gosselin [23]. The effects of passive joint limits and links interference are included in the constant orientation methodology by Merlet [24, 1]. Constant orientation workspaces of other types of six-dof parallel manipulators including the 6-$\underline{P}US$ (Bonev and Ryu [25]) and 6-$\underline{R}US$ (Bonev and Gosselin [26]) parallel manipulators have also been determined by means of the geometrical method.

A hybrid geometrical-numerical method for determining the orientation workspaces of 6-$U\underline{P}S$ parallel manipulators has been proposed by Merlet [19]. This problem is also addressed by Huang *et al.* [27], who propose a closed form solution to the problem.

Bajpai and Roth [28] and Williams and Reinholtz [29] also use geometrical reasoning to determine workspaces of specific classes of manipulators.

Geometrical methods represent the most efficient and accurate methods for workspace determination available, since the workspace boundary is expressed in analytical terms. It is evident however that the various geometric method implementations are specific to distinct classes of manipulators. At present there exist no direct geometric methods for determining maximal and dextrous workspaces of spatial parallel manipulators.

Continuation methods

A broadly applicable method for workspace analysis using continuation methods has been presented by Jo and Haug [30]. In this method, manipulator workspace boundaries are defined as the sets of points for which the Jacobian matrix of the kinematic constraints are row rank deficient. A continuation method is then used to trace the family of one-dimensional trajectories which correspond to the workspace boundary. When determining parallel manipulator workspaces using this methodology, it becomes necessary to use a slack variable formulation to represent the unilateral constraints implied by physical limits to joint motions (Jo and Haug [31]). This is one of the limitations of the continuation method: that the introduction of other constraints limiting the workspace, such as limits on the passive joints, and link interferences, lead to a very large manipulator Jacobian, which in turn may render the procedure difficult to manage (Merlet [1]).

Jo and Haug [31] use the continuation method to determine maximal workspaces of a 3-$R\underline{P}R$ planar manipulator, and constant orientation workspaces of a 6-$S\underline{P}S$ spatial manipulator. Further developments by Haug *et al.* [16] result in the determination of maximal workspaces of 6-$S\underline{P}S$ spatial parallel manipulators. In Haug *et al.* [32] it is shown that the continuation method also provides an effective tool for determining barriers to control within the manipulator workspace.

The determination of dextrous workspaces of 3-$R\underline{P}R$ and 6-$S\underline{P}S$ parallel manipulators (Haug *et al.* [17]) has also been addressed by means of the continuation method. These authors motivate their approach by stating that "numerical methods are required for constructing boundaries of dextrous workspaces [of more complex spatial manipulators]", an assertion that is borne out by the fact that there are at current no analytical methods available for solving this problem.

Continuation methods have also been applied to the problem of determining *operational envelopes*, or the set of points which can be occupied by *all* points on the working body of the manipulator (Haug *et al.* [33], Adkins and Haug [34]). This problem is important in order to avoid interference between the working body and its surroundings. An associated problem is the determination of domains of interference between working bodies and their surroundings (Haug *et al.* [35, 36]).

An overview of continuation methods applied to determination of workspaces, operational envelopes, and domains of interference is given by Haug *et al.* [37].

Discretization methods

Although computationally expensive, discretization methods represent an easy and stable method for workspace determination. A large variety of implementations are found in the literature. One approach (Yang and Lee [38], Sorli and Ceccarelli [39], Cervantes-Sánchez and Rendón-Sánchez [40]) is to vary the manipulator *input* parameters discretely between their limits, and plot the points reached by the reference point. This approach can provide some insight into the effects of design parameters on the manipulator workspace (Ceccarelli and Sorli [41]), but produces results which can be difficult to interpret, and requires the solution of the forward kinematics of the parallel manipulator.

Since the inverse kinematics are easy to solve for a parallel manipulator, seemingly a better approach is to discretize the *output* space of the manipulator, and then determine whether or not each discrete point belongs to the workspace by solving the inverse kinematics and evaluating at that point the various constraints acting on the manipulator. Approaches based on this idea have been used by Fichter [42], Lee and Shah [43], Masory and Wang [44], Arai *et al.* [45], Stamper *et al.* [46] and Wang *et al.* [47] for determining constant orientation and maximal workspaces of various manipulators. Bonev and Ryu [25] propose a discretization method for determining orientation workspaces of 6-$U\underline{P}S$ manipulators.

The main criticism of discretization methods is their exponential increase in computational expense as the required accuracy is increased.

Optimization methods

Of the methods discussed in the previous sections, geometric methods are highly efficient and accurate, but require a specific formulation for each manipulator class and workspace type. At the other extreme discretization methods are computationally intensive resulting in limited accuracy, but can easily be applied to almost any manipulator. Numerical continuation methods lie somewhere between these two approaches, but including all the constraints acting on the manipulator, and the fact that all internal boundary curves are also mapped, can make the method difficult to implement for more complex manipulators.

As an alternative efficient *numerical* approach for workspace determination, optimization methods have been proposed by Snyman *et al.* [48]. The basic philosophy of the optimization approach is to define the workspace boundary in terms of a constrained optimization problem, where the constraints relate to various physical conditions which limit the workspace of the manipulator. A numerical optimization algorithm is used to solve the optimization problem in a number of search directions to obtain a representation of the workspace boundary. There are two specific implementations of the optimization approach, the *ray* method and the *chord* method. These two methods are distinguished from each other by the search geometry used in determining the successive discrete points along the workspace boundary.

The original ray method of Snyman *et al.* [48] determines the points of intersection of a pencil of rays emanating from a fixed radiating point with the workspace boundary. One deficiency of the the ray method is that it cannot be used to map non-convex manipulator workspaces. The modified ray method (Hay and Snyman [49]) addresses this problem by using user interaction together with the original ray method to map sections of the workspace which cannot be mapped automatically. The alternative chord method of Hay and Snyman [50] can be used to determine non-convex manipulator workspaces automatically. From an initial point on the workspace boundary, the chord method uses fixed radius arc searches to determine successive points until closure of the boundary is obtained.

Previous applications of the optimization approach have focussed on the determination of constant orientation (ray methodology - Du Plessis and Snyman [18]) and maximal workspaces (ray method - Snyman *et al.* [48]; modified ray method - Hay and Snyman [49]; chord method - Hay and Snyman [50]) of planar $3\text{-}R\underline{P}R$ and spatial $6\text{-}U\underline{P}S$ manipulators. An efficient indirect method for determining dextrous manipulator workspaces of these same manipulators by calculating the intersection of various constant orientation workspaces has also been proposed by Du Plessis and Snyman [18].

In Section 5.5 of this work a *direct* method for determining dextrous planar manipulator workspaces is given.

In this study, the chord method is used as the method of choice for determining manipulator workspaces when using the optimization approach. A general overview of the chord methodology is given in Appendix C. A detailed presentation and applications of the ray methodology may be found in Snyman *et al.* [48] and Du Plessis [51]. The modified ray and chord methods are further discussed in Hay and Snyman [49, 52] and Hay [53].

An optimization approach similar to the ray method has also been suggested by Wang and Hseih [54].

1.4 Optimal design of parallel manipulators

As already mentioned parallel manipulators possess a number advantages over traditional serial manipulators (Merlet [1]). Parallel manipulators are, however, difficult to design due to their highly nonlinear and often non-intuitive behavior. An effective and systematic way of addressing the problems stated above is through the use of optimization techniques in the design process. Depending on the particular application, certain manipulator performance criteria may be of more importance than others. Such criteria include design so that the manipulator can reach a certain prescribed workspace, design for optimum velocity, force or error transmission ratios between the actuators and the moving platform, stiffness, isotropy, dynamic behavior or dexterity of the manipulator throughout the workspace.

A distinction can be made between the types of problem studied in the current literature in terms of whether or not workspace requirements are included in the optimization. In the next section optimization purely with respect to some performance measure is discussed. The different synthesis problems, which explicitly include requirements on the workspace, are discussed in Section 1.4.2.

1.4.1 Optimization of performance

This first type of problem involves optimizing the *performance* of the manipulator with respect to some performance measure without explicit consideration of the workspace. The various performance measures commonly used in optimizing manipulators are listed below.

Stiffness

Bhattacharya *et al.* [55], investigate the effect of design parameters on the stiffness of a 6-$U\underline{P}S$ manipulator. Since these authors consider only two variables, the optimization is performed by plotting various stiffness measures as functions of the design parameters and then selecting the most appropriate design by inspection of these graphs. A method for synthesizing a manipulator with respect to link and joint stiffnesses so that the end effector has a desired stiffness is suggested by Chakarov [56]. Hayward *et al.* [57] notice the importance of manipulator stiffness in designing a parallel mechanism-based hand controller.

The optimal design of 3-dof spherical manipulator with respect to both stiffness and conditioning is undertaken by Liu *et al.* [58]. Again, since there are only two design parameters, the optimization is done by inspection of plots of these performance measures against the design parameters. Zhang *et al.* [59] use a genetic algorithm to optimize the stiffness of a 5-dof revolute actuated parallel manipulator with passive constraining leg. Simaan and Shoham [60] determine the configurations of a variable geometry 3-$R\underline{P}R$ planar parallel manipulator which yield a desired stiffness of the end-effector.

Static and dynamic behavior

A mechanism is said to be statically balanced when the weights of the links do not produce any torque (or force) at the actuators under static conditions (Gosselin [61]). Such balancing is achieved through the use of counterweights or springs. Static balancing of various spatial parallel mechanisms has been performed analytically by Wang and Gosselin [62, 63], Gosselin and Wang [64] and Gosselin *et al.* [65]. Herder [66] provides a general discussion on statically balanced parallel mechanisms.

Dynamic behavior is mentioned by Merlet [1] as a measure to be used when quantifying manipulator performances. In many practical applications optimization of acceleration and inertial characteristics may be of importance. Weck and Giesler [67] include dynamic properties in their multi-objective optimization of a 2-$R\underline{P}R$ planar machine tool.

Conditioning of the Jacobian matrix and dexterity

The condition number of the Jacobian matrix of the manipulator can be used as a measure of accuracy of control of the manipulator, or manipulator dexterity. Here the term dexterity refers to a measurement of fine end-effector motion in a local sense (Klein and Blaho [68]). When viewed as a measure of accuracy, the condition number can be thought of as a factor amplifying errors in the actuators, and thus affecting the natural precision of the manipulator. The best conditioning is obtained when the Jacobian matrix is orthogonal, and the manipulator is said to be in an isotropic configuration.

Gosselin and Angeles [21, 69] study planar and spherical 3-$\underline{R}RR$ manipulators and determine, amongst other conditions, designs so that these manipulators are isotropic in their home configurations. Gosselin and Lavoie [70] determine designs of spherical 3-dof manipulators so that they have at least one isotropic configuration. Pittens and Podhorodeski [71] and Zanganeh and

Angeles [72] undertake the isotropic design of 6-$U\underline{P}S$ spatial and 3-$\underline{R}RR$ planar manipulators. It is noted by these authors and by Merlet [1] that it is necessary, if the Jacobian contains both rotational and translational terms, to scale the translational terms by a chosen characteristic length, since the Jacobian is not invariant under any choice of dimensional units. This is a criticism of the use of the condition number as a performance measure in these cases (Merlet [73]).

Of course, isotropy of the Jacobian matrix is a *local*, configuration-dependent property of the manipulator. Gosselin and Angeles [74] propose a global conditioning index (GCI), evaluated over the entire workspace of the manipulator, which they used to optimize the *global* conditioning of parallel 3-dof planar and spherical manipulators using the complex method[3]. The GCI is aimed at obtaining better performance of the manipulator throughout its workspace. Stamper *et al.* [46] and Tsai and Joshi [75] use the global condition index to numerically optimize a spatial 3-dof translational parallel manipulator. Kurtz and Hayward [76] and Leguay-Durand and Reboulet [77] use similar principles in optimizing redundantly-actuated spherical mechanisms. In this case since the number of design variables is low the optimization can be performed graphically. Stoughton and Arai [78] argue against averaging the conditioning over the *entire* workspace and propose instead optimizing the average dexterity over a centralized subregion of the workspace of a 6-$U\underline{P}S$ spatial manipulator. The optimization is performed using the numerical BFGS algorithm.

Other

In Lee *et al.* [79, 80], Zhang and Duffy [81] and Lee *et al.* [82] the concept of using the quality index to determine optimal designs and configurations is proposed and developed for various 3-$R\underline{P}R$ and n-$U\underline{P}S$ manipulators. The

[3] The complex method is a constrained version of the simplex method (Box 1965)

quality index is a measure of proximity to a singularity. Carretero *et al.* [83] minimize the parasitic motion of a 3-$\underline{P}RS$ spatial parallel manipulator using a quasi-Newton optimization method.

1.4.2 Workspace synthesis

The second type of problem is concerned with the *workspace* of the manipulator. Essentially this type of problem, concerned with manipulator synthesis, is the inverse of the analysis problem. Analysis is concerned with determining the workspace of the manipulator for a given design and dimensioning. Synthesis seeks to find the dimensions of the manipulator so that is has a required workspace. Since there is not necessarily a unique solution to this problem, additional requirements are sometimes introduced, where required, to ensure a desired performance of the mechanism as well.

Synthesis with respect to workspace only

Gosselin and Guillot [84] use the complex method to optimize a planar 2-$R\underline{P}R$ parallel manipulator so that the workspace of the manipulator is as close as possible to a prescribed workspace. This methodology is extended by Boudreau and Gosselin [85, 86] and is applied, now using a genetic algorithm, to planar 3-$R\underline{P}R$ and 3-$\underline{R}RR$ manipulators, and a spatial 6-$U\underline{P}S$ manipulator.

Murray *et al.* [87] use a quaternion approach which allows them to determine many 3-$R\underline{P}R$ planar manipulator designs, the workspaces of which include a number of prescribed points. In this approach the set of serial chains (forming the links between the base and moving platform) that can reach the desired poses are first determined. The feasible parallel manipulator designs which can also reach these poses are then assembled from these chains. The quaternion approach is extended to other parallel manipulator types, both planar and spatial in Murray and Hanchak [88], and Perez and McCarthy [89].

Another approach proposed by Merlet [90, 91] is used to determine all spatial manipulator geometries, the workspaces of which include prescribed points or line segments. The method presented takes into account leg length constraints, limits on the passive joint angles, and interference between links. The basic approach is to use each constraint separately to restrict the design variable domain. The region where all constraints are satisfied then corresponds to the set of manipulator designs which can reach the desired poses. Merlet [92, 1] later extends the methodology to include constraints on the articular velocities.

The effects of the design parameters on workspaces of various spatial parallel manipulators is also studied by Ji [93].

Multi-objective optimization

The algorithm for workspace synthesis proposed by Merlet [91] has been extended in Merlet [92], resulting in the DEMOCRAT design methodology. Once the design space has been reduced to all the robot designs, which can reach the required workspace as described in the previous section (referred to by Merlet as the "cutting phase"), the "refining phase" then consists of discretizing this reduced design space, and evaluating robot performances with respect to various performance criteria at each resulting node. The advantage of the DEMOCRAT methodology is its ability to determine *all* possible designs which fulfill the designer's requirements. It would appear though, that as the number of design variables increase, the methodology becomes increasingly more difficult to handle.

Kirchner and Neugebauer [94] have combined many performance criteria in optimizing a spatial manipulator with 13 design variables. These authors determine the Pareto-optimal set with respect to these criteria using a genetic algorithm. Similarly Weck and Giesler [67] perform multi-objective optimization of a planar manipulator to be used in machining applications.

Finally, Gallant and Boudreau [95] propose a method which uses a genetic algorithm for synthesizing planar parallel 3-$R\underline{P}R$ manipulators for a desired workspace, including avoidance of singularities and using the global conditioning index.

1.5 Numerical optimization methods

The sustained increase in computing power has led to numerical optimization techniques becoming more and more popular in many fields, including mechanical engineering. Most current optimization algorithms can be broadly classified as either *deterministic* or *stochastic* methods (Chedmail [96]). The methods discussed in this section are in general for nonlinear optimization problems, since these are the sorts of problems which occur in the mechanism synthesis field.

1.5.1 Deterministic methods

Deterministic methods use knowledge of the local topography of the objective (and constraint) function to travel towards the optimum design. Such methods include classical optimization techniques, line search methods, gradient-based methods and methods such as the simplex method and the method of moving asymptotes. Although many such methods exist, the discussion here is limited to methods which have direct relevance to this study.

Line search methods

The method of *steepest descent* is one of the most fundamental procedures for minimizing a differentiable function of several variables. The method, proposed by Cauchy in the middle of the nineteenth century, continues to be the basis of several gradient-based solution procedures (Bazaraa *et al.* [97], p.300). The performance of the steepest descent method is disappointing, however, compared to other first-order (gradient only) line search methods. In spite of using what appears to be the "best" search direction, i.e. that which gives the maximum rate of decrease at the point of application, the method is not really effective in most problems. The method of steepest descent usually works quite well during the early stages of the optimization process, depending on the point of initialization. However, as a stationary point is approached, the method often behaves poorly, taking small and nearly orthogonal steps. Steepest descent methods are discussed more fully in Section 2.2.

Amongst the methods that use only gradient information and perform successive line searches, the most popular method is probably the *conjugate gradient* method of Fletcher and Reeves [98]. This method generates mutually conjugate directions by taking, at each successive point, a suitable convex combination of the current gradient and the direction used at the previous iteration, as search direction. A slight variation of the Fletcher-Reeves method is the method of Polak and Ribière [99], which is argued to be preferable for non-quadratic functions (Bazaraa *et al.* [97], p.357). Gradient only methods, such as the Fletcher-Reeves method, remain of great importance because they become indispensable when the problem size (number of variables) becomes very large.

Second order methods, using Hessian information and based on Newton's method, have also been proposed. For large numbers of variables the full evaluation of the Hessian matrix, required by Newton's method at each step, becomes time-consuming. In order to avoid this difficulty, *quasi-Newton* methods, which approximate the Hessian matrix by means of an update formula after each step, have been proposed. Two implementations of such quasi-Newton methods are the Davidon-Fletcher-Powell (DFP), and the more recent Broyden-Fletcher-Goldfarb-Shanno (BFGS) methods. These methods exhibit fast convergence. When the number of variables exceed approximately 100, however, attempts to update the Hessian become impractical because of the size of the matrix (Bazaraa *et al.* [97], p.328).

Lagrangian-based methods

In 1760 Lagrange developed the *classical* method for solving equality constrained optimization problems by the introduction of Lagrange multipliers and solution of the resulting unconstrained optimization problem (Snyman [100]). This method can be extended to inequality constrained problems by the use of auxiliary variables. Karush (1939) and Kuhn and Tucker (1951) derived the necessary Karush-Kuhn-Tucker (KKT) conditions, expressed in terms of the Lagrangian, that must be satisfied at the solution of an inequality constrained problem. Analytical determination of the stationary point of the Lagrangian is not always practical, or possible. In order to address this problem, *augmented Lagrange multiplier* methods combine the classical Lagrangian method with a penalty function approach in solving the constrained problem. Here successive approximations to the Lagrange multipliers are used in order to obtain the solution via an iterative procedure.

Sequential quadratic programming (SQP) methods are based on the application of Newton's method to determine the optimum from the KKT conditions of the constrained problem. The method relies on the solution of a quadratic programming problem at each step, in order to determine the next approximate solution and associated Lagrange multipliers. A complete discussion of SQP methods is given in Section 3.2.

The dynamic trajectory method

An alternative optimization algorithm, based on modelling the dynamic trajectory followed by a particle of unit mass in a conservative force field has been proposed by Snyman [101, 102, 103]. This method, called the dynamic trajectory, or Leap-Frog (LfopC) optimization algorithm, has a number of properties which make it suitable for implementation in solving practical engineering optimization problems. A detailed discussion of this method appears in Appendix B.

1.5.2 Stochastic methods

In contrast to deterministic methods, stochastic methods only use gradient information indirectly, and use instead *random* processes for finding new points in the design space. Such methods usually rely on modelling natural phenomena as the basis of the algorithm. Examples of this type of method are the genetic, simulated annealing and particle swarm algorithms.

Genetic algorithms

Genetic algorithms were first introduced by Holland (1965). Their use has subsequently been encouraged by Goldberg [104] and Michalewicz [105]. These algorithms mimic the process of evolution found in nature. From an initial, random population, where each individual is characterized by a specific design vector, subsequent generations are created by "inheriting" features, or parts of the design vector, from the previous generation. The best individuals in each generation are given a better chance of passing on their features to subsequent generations, thus driving the entire population towards an optimum design. Various strategies, such as introducing random perturbations or "mutations" into the design vector of certain individuals, are also employed. Genetic algorithms have gained tremendous popularity due to their high ease of programming, and ability to take into account discrete and continuous design variables (Chedmail [96]). These algorithms are however computationally very demanding, and often require the experimental selection of many optimization parameters in order to obtain good performance for a given problem.

Simulated annealing

Optimization by simulated annealing was proposed by Kirkpatrick *et al.* [106] who credit Metropolis *et al.* [107] for the basic idea. The problem studied by Metropolis and his colleagues was to determine the equilibrium state of a material, composed of a number of particles, by simulating the thermal motion of these particles at a given temperature. In order to use this simulation as a component in an optimization technique, the temperature is used as a control parameter, and under systematic reduction of this, the algorithm asymptotically and statistically converges to the global optimum of the system being optimized. The difficulty associated with such algorithms is that the efficiency of the algorithm, and accuracy of results, are affected by the choice of parameters, such as the rate of decreasing the control temperature. As with genetic algorithms, some initial experimentation is necessary to determine the best settings for a given problem. Many function evaluations are also required in comparison to deterministic methods.

1.6 Motivation for the study

1.6.1 Optimization of parallel manipulators

Some of the advantages offered by parallel manipulators, when *properly designed*, include an excellent load to weight ratio, high stiffness and positioning accuracy and good dynamic behavior (Merlet [2, 1], Fichter and MacDowell [8]). These characteristics are, to a large extent, the result of the load on the platform being distributed more or less equally among the actuators, as opposed to the serial case where the full load is carried by each actuator. In addition, depending on the exact parallel manipulator design, the stress in the actuators is mostly tension or compression, which means that the manipulator can be made very rigid, especially when using linear actuators. In contrast, for a serial manipulator the load is often carried in a cantilever fashion, and the mechanism must be designed to carry the resulting bending loads, often resulting in bulky links. Another factor influencing the accuracy of parallel manipulators is that the positioning accuracy of the end-effector is only slightly affected by errors in the actuators. Errors tend to average in the parallel case, whereas they are cumulative for a serial robot. All of these factors, and the availability of new control and component technologies, have resulted in the increasing popularity of parallel manipulators.

There are, however, also some disadvantages associated with parallel manipulators, which have inhibited their application in some cases. Most serious of these is that the particular architecture of parallel manipulators leads to smaller manipulator workspaces than their serial counterparts. This is due to the additional constraints imposed by the closed kinematic chains of such mechanisms. Parallel manipulators can also be difficult to design (Gosselin *et al.* [108]), since the relationships between design parameters and the workspace, and behavior of the manipulator throughout the workspace, are not intuitive by any means. In addition, parallel manipulator performances are highly dependent on their dimensions. Merlet [73] gives the example that changing the radius of a Gough-Stewart platform by 10% results in a 700% change in the minimal stiffness of the robot. For all of these reasons, Merlet [1] argues that customization of parallel manipulators for each application is absolutely necessary in order to ensure that all performance requirements can be met by the manipulator.

1.6.2 The need for new methodologies

In their recent review paper Dasgupta and Mruthyunjaya [3] survey 214 relevant publications, and at the end of the paper state that, amongst others, the following open problems exist:

1. A detailed and easy-to-use description of the workspace.
2. Workspace synthesis for the Stewart platform.
3. Optimum kinematic synthesis of the Stewart platform for well-conditioned workspace.

These notions are supported even more recently by Merlet, who devotes a keynote address to "the need for a systematic methodology for the evaluation and optimal design of parallel robots" [109]. In a later paper the same author [110] states that "none of [the existing dimensional synthesis methods] are appropriate for parallel robots, which usually have a large number of design parameters".

Of the synthesis methods discussed in Section 1.4.2, genetic algorithm approaches are capable of synthesizing manipulators with large numbers of design variables. These methods are however disadvantaged by their reliance on weighting the contributions of individual performance measures when performing the multi-objective optimization, and the *high computational expense* of these optimization algorithms. Alternative approaches, while efficient, are limited by their need to derive a specific formulation for various manipulator types. In addition, increasing the number of manipulator variables leads to dramatically increasing complexity of the methods. It is thus felt that there is a need for a design methodology, based on *efficient* numerical optimization techniques, which is generally applicable to a variety of manipulator architectures.

1.6.3 Objectives of this study

In an effort to address the points highlighted in the previous section, the following issues are addressed in this study:

1. The development of efficient numerical optimization algorithms capable of handling engineering problems.

2. The development and refinement of numerical methods for workspace determination of various parallel manipulators.

3. The development of alternative numerical methods for manipulator dimensional synthesis, and the investigation of the applicability of the new optimization algorithms, mentioned in 1, to such problems.

This work is split into two parts:

PART I: OPTIMIZATION ALGORITHMS is devoted to the development of new optimization algorithms (item 1 above). Two separate numerical optimization algorithms are presented. The spherical quadratic steepest descent (SQSD) algorithm, presented in **Chapter 2** is intended for unconstrained problems. In **Chapter 3** an optimization algorithm for constrained problems, called the Dynamic-Q algorithm, is presented.

PART II: MANIPULATOR OPTIMIZATION is devoted to workspace determination and development of new methods for manipulator optimization (items 2 and 3 above). In **Chapter 4** various optimization algorithms are applied to the problem of synthesizing a 2-$R\underline{P}R$ planar parallel manipulator. Various forms of the optimization problem statement are developed and evaluated. Building on these results, **Chapter 5** contains application of the methodology to a planar 3-$R\underline{P}R$ manipulator, together with some new developments for dextrous workspace determination of such manipulators. A different class of manipulator, tendon-driven parallel manipulators, are studied in **Chapter 6**. New analysis methods for this class of manipulator are introduced, and optimization of such manipulators is performed. Finally in **Chapter 7** conclusions are drawn from the work performed, and recommendations for future research are made.