Part II

MANIPULATOR OPTIMIZATION

Chapter 4

Choice of a suitable dimensional synthesis methodology

4.1 Introduction

The main objective of this chapter is to find a fundamentally sound and robust *numerical* methodology for synthesizing parallel manipulators for a desired workspace. Various formulations for manipulator dimensional synthesis are proposed and investigated numerically by application to a two-degree-of-freedom (dof) parallel manipulator. Specifically, the various synthesis strategies result in one unconstrained optimization formulation and two constrained optimization formulations. In the next section, the 2-$R\underline{P}R$ planar parallel manipulator studied in this chapter is described. Thereafter four candidate numerical optimization algorithms, LfopC, EtopC, Dynamic-Q and SQSD, are briefly discussed. These four optimization algorithms are applied to the the unconstrained O synthesis (definition follows below) manipulator design formulation in order to assess their relative merits and po-

tential for application to parallel manipulator dimensional synthesis problems in general. The goal of the O synthesis formulation is to determine a manipulator design so that its workspace is as close as possible to some prescribed workspace. The most suitable optimization algorithm identified by means of the O synthesis problem is then applied to the proposed (constrained) E and P synthesis formulations. The E synthesis formulation seeks to determine the manipulator design so that workspace of the optimal manipulator fully contains a prescribed workspace in the most efficient manner. The second, more practical, P synthesis formulation is aimed at determining a manipulator design so that a prescribed workspace is fully enclosed, but that the workspace is also well-conditioned with respect to some performance index. The final methodology arrived upon is applied to the more complicated case of a 3-dof planar parallel manipulator in the next chapter.

4.2 Coordinates and kinematic constraints

In general terms, when describing the kinematics of a mechanism, the following descriptions and definitions can be used (Haug *et al.* [16]). *Generalized coordinates* $\mathbf{q} = [q_1, q_2, \ldots, q_{nq}]^\top \in \Re^{nq}$ are used to characterize the position and orientation of each body in the mechanism. It the vicinity of an *assembled configuration* of the mechanism, these generalized coordinates satisfy m independent holonomic kinematic constraint equations of the form

$$\mathbf{\Phi}(\mathbf{q}) = \mathbf{0} \tag{4.1}$$

where $\mathbf{\Phi} : \Re^{nq} \rightarrow \Re^{m}$ is a smooth function.

Mechanisms are usually designed to produce a desired functionality. Specifying the values of a selected subset of the generalized coordinates, called the *input coordinates*, defines the motion of the mechanism. These input coordinate values are controlled by external influences with the intent of prescribing the motion of the mechanism. The vector of input coordinates is

denoted by $\mathbf{v} = [v_1, v_2, \ldots, v_{nv}]^\mathsf{T} \in \Re^{nv}$.

In order to characterize the functionality of the mechanism some measure of output, which is controlled by the mechanism inputs, must be monitored. *Output coordinates* are the subset of the mechanism's generalized coordinates that define the useful functionality of the mechanism. Output coordinates are distinct from input coordinates and are denoted by $\mathbf{u} = [u_1, u_2, \ldots, u_{nu}]^\mathsf{T} \in \Re^{nu}$. A choice of input and output coordinates for a mechanism defines a mechanical system with an intended function. This mechanism is then called a manipulator.

Generalized coordinates of a mechanism that are neither input coordinates nor output coordinates are called *intermediate coordinates*, denoted by $\mathbf{w} = [w_1, w_2, \ldots, w_{nw}]^\mathsf{T} \in \Re^{nw}$, where $nw = nq - nv - nu$.

4.3 The planar two-degree-of-freedom parallel manipulator

The mechanism used to investigate the various methodologies proposed in this chapter is a planar 2-$R\underline{P}R$ parallel manipulator similar to that studied by Gosselin and Guillot [84]. As shown in Figure 4.1, the manipulator consists of two linear actuators, with variable lengths l_1 and l_2, connected to the ground by means of revolute joints A and B, and to each other by a revolute joint P with global coordinates (x_P, y_P). Point A has coordinates (x_A, y_A) and point B coordinates (x_B, y_B). It assumed here that $y_B = y_A$. Point P will be used to describe the motion of the mechanism and is called the *working point* of the manipulator. Here for the sample 2-dof manipulator, with no limits on the actuator lengths, point P may arbitrarily be positioned in the $x-y$ plane by controlling the lengths of legs 1 and 2. It is evident that this manipulator thus has two degrees of freedom. The *input* coordinates, which are used to control the manipulator, are the leg lengths $\mathbf{v} = [l_1, l_2]^\mathsf{T} \in \Re^2$. *Output* coordinates,

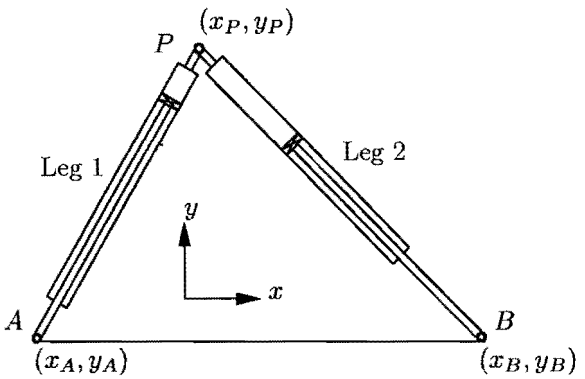

Figure 4.1: The 2-dof parallel manipulator

describing the functionality of the mechanism, are $\mathbf{u} = [x_P, y_P]^\mathsf{T} \in \Re^2$. There are no intermediate coordinates. Thus for this 2-dof example $nu = 2, nv = 2$ and $nw = 0$. The generalized coordinates of the manipulator are partitioned as follows

$$\mathbf{q} = [\mathbf{u}^\mathsf{T}, \mathbf{v}^\mathsf{T}]^\mathsf{T} \in \Re^4 \tag{4.2}$$

For this two-degree-of-freedom manipulator there are two kinematic constraint equations (i.e. $m = 2$), and (4.1) can be rewritten in terms of the partitioning given by (4.2)

$$\mathbf{\Phi}(\mathbf{q}) = \mathbf{\Phi}(\mathbf{u}, \mathbf{v}) = \mathbf{0} \tag{4.3}$$

The motion of the manipulator is restricted when the actuator legs have limits associated with them of the form

$$l_i^{\min} \leq l_i \leq l_i^{\max}, i = 1, 2 \tag{4.4}$$

or more generally

$$\mathbf{v}^{\min} \leq \mathbf{v} \leq \mathbf{v}^{\max} \tag{4.5}$$

where $\mathbf{v}^{\min} = [l_1^{\min}, l_2^{\min}]^\mathsf{T}$ and $\mathbf{v}^{\max} = [l_1^{\max}, l_2^{\max}]^\mathsf{T}$.

These constraints, together with the geometry of the manipulator, determine the size and shape of the workspace of the manipulator. Since the

manipulator has no orientational ability, the types of workspace discussed in Section 1.3.1 have no meaning here, and the workspace W is simply the set of points that can be reached by the working point P of the manipulator. The workspace can thus be defined as

$$W = \{\mathbf{u} \in \Re^{nu} : \mathbf{\Phi}(\mathbf{u}, \mathbf{v}) = \mathbf{0}, \text{ with } \mathbf{v} \text{ satisfying (4.5)}\} \tag{4.6}$$

The boundary ∂W of the workspace may then be defined as

$$\begin{aligned} \partial W = \{&\mathbf{u} \in \Re^{nu} : \mathbf{u} \in W \text{ and } \exists \text{ an } \mathbf{s} \in \Re^{nu} \text{ such that for} \\ &\mathbf{u}' = \mathbf{u} + \lambda \mathbf{s}, \lambda \in \Re \text{ arbitrarily small and either} \\ &\text{positive or negative, no } \mathbf{v} \text{ exists that satisfies} \\ &\mathbf{\Phi}(\mathbf{u}', \mathbf{v}) = \mathbf{0} \text{ as well as inequalities (4.5)}\} \end{aligned} \tag{4.7}$$

The particular workspace determination method used here is the chord optimization method (Snyman and Hay [122]) although the ray optimization method (Snyman *et al.* [48]) would be equally applicable. The chord method is fully described in Appendix C and yields on application discrete points $\mathbf{b}^i \in \mathbf{u}, i = 1, 2, \ldots, n_b$ along the boundary of the workspace at constant chord lengths d, as shown in Figure 4.2. Included in this set of points are the bifurcation points $\mathbf{B}^j \in \mathbf{u}, j = 1, 2, \ldots, N_B$. It is arbitrarily assumed that all points are ordered counterclockwise.

4.4 Candidate optimization algorithms

In this preliminary study four different optimization algorithms, all developed at the University of Pretoria, were compared in order to assess their suitability for solving the manipulator optimization problems to be addressed. The four candidate algorithms were the unconstrained spherical quadratic steepest descent optimization algorithm, and three constrained optimization algorithms: LfopC, Dynamic-Q and EtopC.

Of these algorithms, the SQSD method, developed in Chapter 2, has proven to work particularly well on ill-conditioned problems with large numbers of variables, however the fact that this is an unconstrained algorithm limits its suitability for manipulator optimization problems. The LfopC algorithm ((Snyman [103], see Appendix B)) possesses a number of outstanding characteristics making it a good potential candidate for implementation. The algorithm requires only gradient information, and no explicit line searches or function evaluations are performed. These properties, together with the influence of the fundamental physical principles underlying the method, ensure that the algorithm is extremely robust. This has been proven over many years of testing (Snyman [103]). A further desirable characteristic related to its robustness, is that if there is no feasible solution to the problem, the LfopC algorithm will still find the best possible compromised solution without breaking down. The one disadvantage of the LfopC method is that, although it moves quickly to the vicinity of the solution, it may converge relatively slowly towards the exact optimum if high accuracy is required. In an attempt to reduce the number of gradient evaluations required, while still retaining the robust properties of LfopC, the Dynamic-Q method (Chapter 3) may prove to be useful.

In order to obtain a gradient-only method with fast convergence in the vicinity of the solution, the conjugate gradient method has been adapted to require only gradient information (Snyman [114]). In the resultant algorithm EtopC, the use of only gradient evaluations is made possible by exploiting a step size selection procedure based on an Euler-trapezium integration scheme. Either the Fletcher-Reeves or Polak-Ribiere directions can be used. In this study, the Polak-Ribiere version of EtopC has been used.

4.5 Best overall fit to a prescribed workspace

The goal of the first formulation presented in this section is stated as

> **O synthesis:** *Determine a manipulator design that results in a workspace W_c which most closely approximates a prescribed workspace W_p.*[1]

4.5.1 Optimization formulation

As indicated in Figure 4.2, the prescribed and calculated workspace boundaries are defined using polar coordinates (β, r), centered on a local coordinate system $x' - y'$ at $\mathbf{O}'$. The angle between the x'-axis of the local coordinate system and the ray to a point $\mathbf{b}_p$ on the prescribed boundary is denoted β_p. The distance from $\mathbf{O}'$ to $\mathbf{b}_p$ is r_p. The boundary of the workspace W_c, associated with specific manipulator design $\mathbf{d}$, may be defined in a similar manner. Here the boundary point $\mathbf{b}_c^i$, generated by the chord method (Snyman and Hay [122], see Appendix C), corresponds to angle β_c^i and ray length r_c^i. For convenience, a point on the prescribed boundary at the angle β_c^i that corresponds to the computed workspace boundary point $\mathbf{b}_c^i$ is denoted $\mathbf{b}_p^i$ and $r_p(\beta_c^i) = r_p^i$.

The part of workspace W_p not intersecting W_c is denoted δW_p, and the part of workspace W_c not intersecting W_p is denoted δW_c (respectively indicated by the light and dark shaded areas in Figure 4.2). It is assumed that the workspace W_c, dependent on the design vector $\mathbf{d}$, will most closely approximate the prescribed workspace W_p when $\mathbf{d}$ is chosen such that the sum of the non-intersecting areas δW_p and δW_c is minimized (Gosselin and Guillot [84]). The optimum solution $\mathbf{d}^*$ is obtained by solving the unconstrained optimization problem :

$$\min_{\mathbf{d}} f(\mathbf{d}) = w_p \delta W_p + w_c \delta W_c \tag{4.8}$$

where the respective weights w_p and w_c in the objective function satisfy the

[1] Subscripts p and c respectively denote quantities associated with the *prescribed* workspace, and workspace *calculated* for a specific design

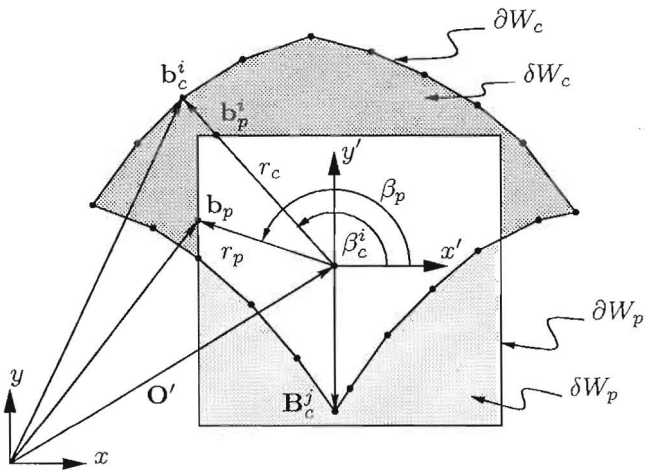

Figure 4.2: Prescribed and calculated workspaces

condition $w_p + w_c = 1$, and are chosen here to be equal, and thus each take on the value 0.5.

The calculation of approximations to the areas δW_p and δW_c is performed using a numerical scheme. Considering two consecutive calculated workspace boundary points $\mathbf{b}_c^i$ and $\mathbf{b}_c^{i+1}$, the incremental contributions dW_p and dW_c to areas δW_p and δW_c may be calculated by use of the following expression, relating the area A of a triangle to the coordinates $\mathbf{L} = [L_x, L_y]^\top$, $\mathbf{M} = [M_x, M_y]^\top$ and $\mathbf{N} = [N_x, N_y]^\top$ of its vertices:

$$A(\mathbf{L}, \mathbf{M}, \mathbf{N}) = \frac{1}{2} \left| M_x N_y + N_x L_y + L_x M_y - M_x L_y - L_x N_y - N_x M_y \right| \quad (4.9)$$

In applying formula (4.9), a distinction must be made between four possibilities that may arise regarding the relative position of the prescribed workspace boundary to any two adjacent calculated workspace boundary points. With

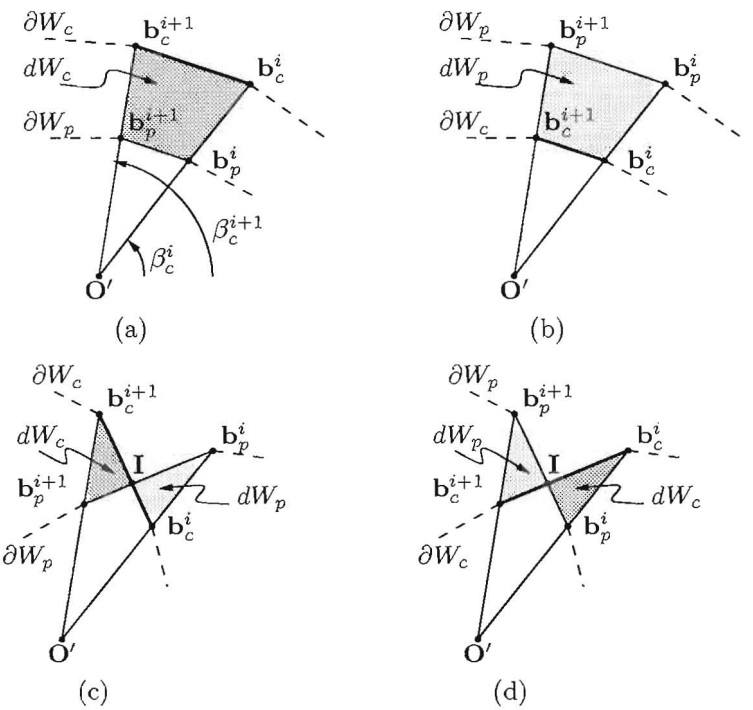

Figure 4.3: Various cases for numerical calculation of workspace areas

reference to Figure 4.3 these four possibilities are:

$$
\begin{aligned}
&\text{a. If } r_c^i > r_p^i \text{ and } r_c^{i+1} > r_p^{i+1} \text{ then} \\
&\qquad dW_c = A(\mathbf{b}_c^i, \mathbf{b}_c^{i+1}, \mathbf{O}') - A(\mathbf{b}_p^i, \mathbf{b}_p^{i+1}, \mathbf{O}'_p) \\
&\text{b. If } r_c^i < r_p^i \text{ and } r_c^{i+1} < r_p^{i+1} \text{ then} \\
&\qquad dW_p = A(\mathbf{b}_p^i, \mathbf{b}_p^{i+1}, \mathbf{O}') - A(\mathbf{b}_c^i, \mathbf{b}_c^{i+1}, \mathbf{O}') \\
&\text{c. If } r_c^i < r_p^i \text{ and } r_c^{i+1} > r_p^{i+1} \text{ then} \\
&\qquad dW_c = A(\mathbf{b}_c^{i+1}, \mathbf{b}_p^{i+1}, \mathbf{I}) \\
&\qquad dW_p = A(\mathbf{b}_c^i, \mathbf{b}_p^i, \mathbf{I}) \\
&\text{d. If } r_c^i > r_p^i \text{ and } r_c^{i+1} < r_p^{i+1} \text{ then} \\
&\qquad dW_c = A(\mathbf{b}_p^i, \mathbf{b}_c^i, \mathbf{I}) \\
&\qquad dW_p = A(\mathbf{b}_p^{i+1}, \mathbf{b}_c^{i+1}, \mathbf{I})
\end{aligned}
\tag{4.10}
$$

where point $\mathbf{I}$ is calculated as required.

Areas δW_p and δW_c can now be approximated by summing the incremental contributions dW_p and dW_c for sequential pairs of boundary points $(\mathbf{b}_c^i, \mathbf{b}_c^{i+1}), i = 1, \ldots, n_{bc} - 2$ where n_{bc} is the number of boundary points generated by the chord method. The closing boundary section is calculated by considering points $(\mathbf{b}_c^{n_{bc}-1}, \mathbf{b}_c^1)$. Note that the final boundary point $\mathbf{b}^{n_{bc}}$ fulfills no part in the calculation of the workspace area since it lies between points $\mathbf{b}_c^1$ and $\mathbf{b}_c^2$. Using this methodology, the objective function in optimization problem (4.8) can be computed for any design vector $\mathbf{d}$. Hence, solving minimization problem (4.8) by means of a suitable optimization algorithm results in a manipulator design $\mathbf{d}^*$ which is a solution to the O synthesis problem.

4.5.2 Numerical results

The four optimization algorithms, listed in Section 4.4, are compared and evaluated in this section to determine which one most efficiently and robustly solves optimization problems of the form (4.8). The particular prescribed workspace chosen is that already considered by Gosselin and Guilliot [84] and shown in Figure 4.4. The choice of this prescribed workspace provides some means of validation of the current results.

The boundary of the prescribed workspace is defined in polar coordinates relative to a local coordinate system centered at $\mathbf{O}' = [0, 3]^\mathsf{T}$:

$$r_p(\beta_p) = \alpha + \sqrt{D^2 \cos^2(\beta_p) + R^2 - D^2} \tag{4.11}$$

where $R = 2.86\dot{7}$ and $D = 1.68\dot{7}$ and the expression for α for various intervals of β_p is given in Table 4.1.

Each of the four candidate optimization algorithms is applied to problem (4.8) using the prescribed workspace defined by (4.11), and run from four different starting designs $\mathbf{d}^0$, given in Table 4.2. The workspaces corresponding to these starting designs are shown, together with the prescribed workspace, in Figure 4.4. A chord length of $d = 0.5$ was used for calculating the

$[\beta_{p\,\min}, \beta_{p\,\max})$	α
$[-\pi/4, \pi/4)$	$-D\cos(\beta_p)$
$[\pi/4, 3\pi/4)$	$-D\sin(\beta_p)$
$[3\pi/4, 5\pi/4)$	$+D\cos(\beta_p)$
$[5\pi/4, 7\pi/4)$	$+D\sin(\beta_p)$

Table 4.1: Parameters specifying the prescribed workspace for O synthesis

	x_A	y_A	x_B	$l_1^{\min}$	$l_2^{\min}$
SP1	-2	-0.075	2	3	3
SP2	-4	-1	4	5	5
SP3	-1	-1	4	3	5
SP4	-5	-1	1	5	3

Table 4.2: O synthesis starting designs

workspace boundary, usually resulting in $n_{bc} \approx 22$ at the solution.

For the 2-dof manipulator there are in total eight parameters which have an effect on the shape and position of the workspace. These parameters are

$$\mathbf{d} = [x_A, y_A, x_B, y_B, l_1^{\min}, l_1^{\max}, l_2^{\min}, l_2^{\max}]^\top \tag{4.12}$$

It is assumed for this illustrative example, in which arbitrary units of length are used, that $y_B = y_A$ and $l_i^{\max} = 1.5 l_i^{\min}, i = 1, 2$. The design vector now becomes

$$\mathbf{d} = [x_A, y_A, x_B, l_1^{\min}, l_2^{\min}]^\top \tag{4.13}$$

For each starting design and algorithm, Tables 4.3 to 4.6 give the number of gradient vector evaluations (N^g) required for convergence, as well as final objective function values $f(\mathbf{d}^*)$ and solution vectors $\mathbf{d}^*$. For all algorithms, and where applicable, the termination tolerances on the function value, step size and gradient norm were respectively set to $\varepsilon_f = 10^{-6}$, $\varepsilon_x = 10^{-4}$ and

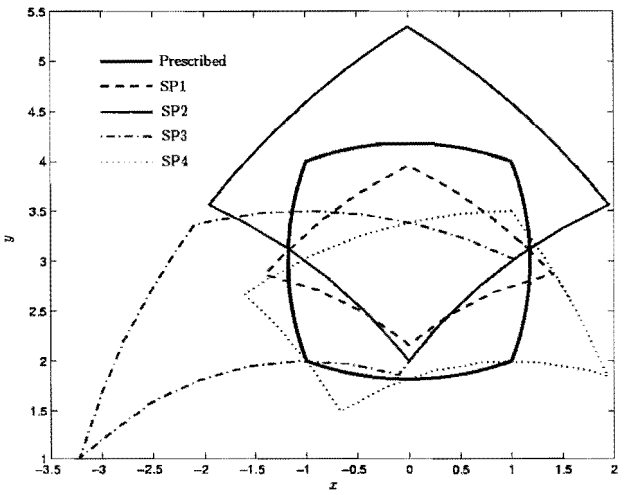

Figure 4.4: Prescribed workspace and workspaces corresponding to starting designs

$\varepsilon_g = 10^{-3}$. Where applicable, move limits were set to $\rho = 0.2$. Figures 4.5 to 4.8 give the convergence histories corresponding to Tables 4.3 to 4.6.

The number of function evaluations (N^f) in the case of the Complex method used by Gosselin and Guillot [84] is not reported. For the Pretoria algorithms (see Section 4.4), because forward finite differences are used for computing the components of the gradient vector, $N^f = (n+1)N^g$, where $n = 5$ is the number of design variables. The interval used in calculating the forward finite differences is $\Gamma = 10^{-5}$ for all the variables.

Algorithm	N^g	f^*	$\mathbf{d}^*$
Dynamic-Q	97	0.596617	$[-3.835, -0.6972, 3.382, 4.277, 4.059]^T$
SQSD	124	0.595935	$[-3.767, -0.6440, 3.552, 4.214, 4.119]^T$
LfopC	152	0.611484	$[-3.962, -0.7344, 3.298, 4.361, 4.058]^T$
EtopC	379	0.611510	$[-3.939, -0.7344, 3.325, 4.348, 4.072]^T$
Complex [84]	-	0.654	$[-4.25, -0.81, 3.02, 4.57, 3.92]^T$

Table 4.3: O synthesis solutions obtained from SP1

Algorithm	N^g	f^*	$\mathbf{d}^*$
Dynamic-Q	32	0.618382	$[-3.873, -0.9893, 3.611, 4.482, 4.324]^\top$
SQSD	90	0.612084	$[-3.890, -0.7318, 3.410, 4.318, 4.114]^\top$
LfopC	161	0.617327	$[-3.947, -0.9825, 3.581, 4.519, 4.309]^\top$
EtopC	444	0.617914	$[-3.941, -0.9638, 3.642, 4.508, 4.326]^\top$

Table 4.4: O synthesis solutions obtained from SP2

Algorithm	N^g	f^*	$\mathbf{d}^*$
Dynamic-Q	80	0.640436	$[-2.928, -0.6551, 4.189, 3.810, 4.502]^\top$
SQSD	71	0.627223	$[-2.818, -0.5753, 4.367, 3.711, 4.540]^\top$
LfopC	186	0.625862	$[-2.896, -0.5905, 4.241, 3.756, 4.472]^\top$
EtopC	228	0.647717	$[-3.005, -0.9524, 3.980, 4.016, 4.501]^\top$

Table 4.5: O synthesis solutions obtained from SP3

Algorithm	N^g	f^*	$\mathbf{d}^*$
Dynamic-Q	86	0.633575	$[-4.075, -0.5207, 2.786, 4.343, 3.679]^\top$
SQSD	189	0.632303	$[-4.005, -0.5264, 2.839, 4.305, 3.707]^\top$
LfopC	116	0.631643	$[-3.878, -0.5418, 2.916, 4.242, 3.753]^\top$
EtopC	509	0.595853	$[-3.828, -0.6684, 3.455, 4.259, 4.081]^\top$

Table 4.6: O synthesis solutions obtained from SP4

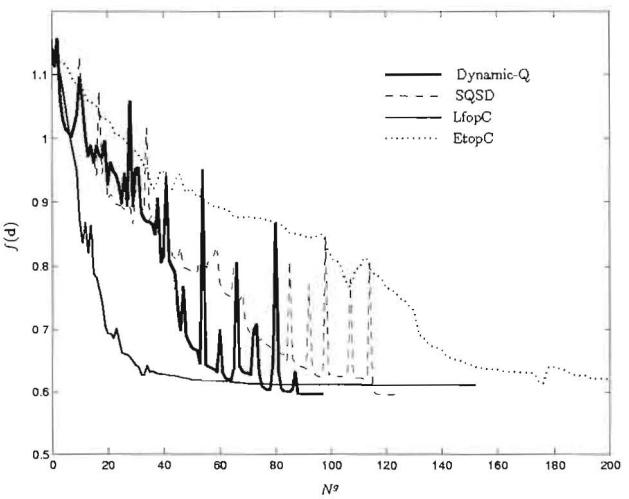

Figure 4.5: O synthesis convergence histories from SP1

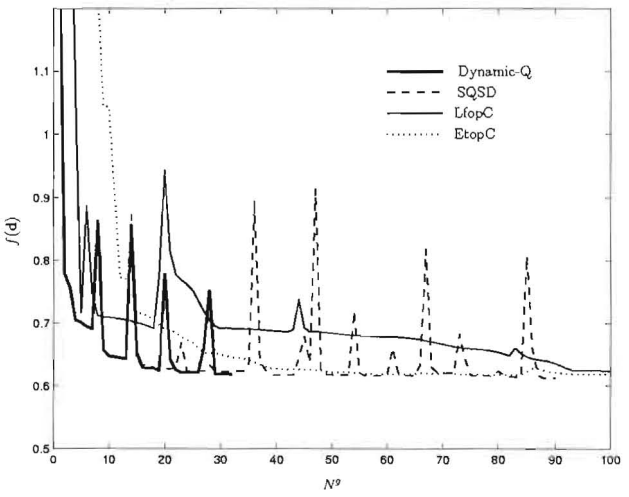

Figure 4.6: O synthesis convergence histories from SP2

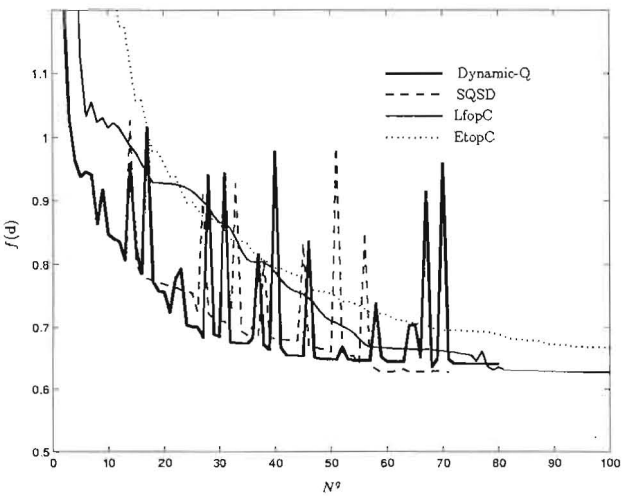

Figure 4.7: O synthesis convergence histories from SP3

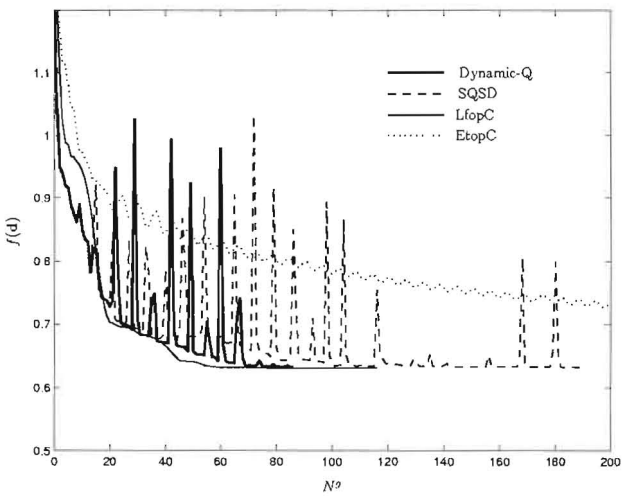

Figure 4.8: O synthesis convergence histories from SP4

In all cases except from SP3, Dynamic-Q is the fastest of the four methods. For SP3, SQSD is marginally faster. As expected, LfopC moves quickly to the vicinity of the solution but takes relatively longer to converge to the accuracy specified. EtopC is the slowest of the four methods but does, however, yield the overall lowest function value of $f^* = 0.595853$ for SP4.

A point of interest relating to the accuracy of the solutions obtained for the O synthesis problem is that the objective function is relatively insensitive to large changes in the design vector. Consider for example the highest (EtopC for SP3) and lowest (EtopC for SP4) solutions obtained. The distance (ℓ_2 norm) between the two solution vectors is 1.127 whereas the difference in function value is only 0.052. This result is typical, and further inspection of the results would suggest a design space which contains a flat-bedded, steep sided solution valley where a large number of different and widely separated design vectors give objective function values close to the global optimum value. Indeed, there does not seem to be a sharp global optimum, or for that matter, sharp local optima since of the 16 runs none converged to identical solutions. A further factor contributing to this is the small amount of numerical noise present in the optimization due to the numerical approximation of areas ∂W_c and ∂W_p.

From a practical point of view it is important to note that all the solutions obtained are almost equally valid in giving designs that closely match the prescribed workspace. This means that any of the algorithms (with indeed any starting point) may in practice be used to satisfactorily solve the design problem. This is further illustrated by Figure 4.9 which depicts the best and worst computed workspaces relative to the prescribed workspace. For comparative purposes the Complex Method solution of Gosselin and Guillot [84] is also shown.

The only remaining consideration is that of computational expense. Although the Dynamic-Q algorithm does not find the lowest solution for each starting point, the solutions found are very good, and its superior perfor-

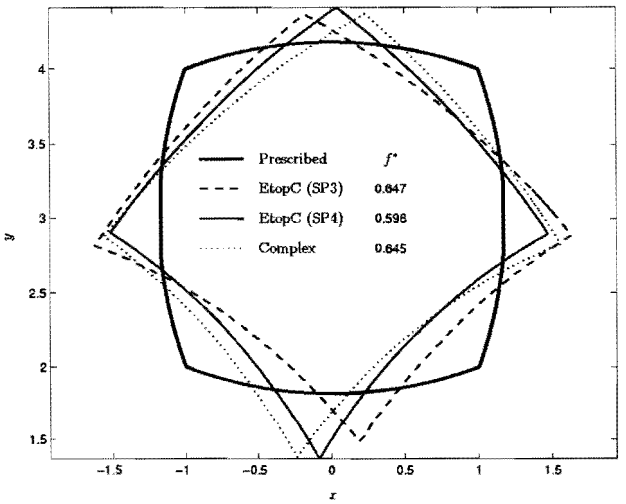

Figure 4.9: O synthesis workspaces corresponding to optimal designs

mance with respect to computational efficiency offsets any other slight disadvantage. Although based on the same approximation principles as Dynamic-Q, SQSD appears to be less efficient, requiring overall many more gradient evaluations than Dynamic-Q. The difference in performance between the two methods can be attributed to the fact that, in SQSD, the constructed approximations are solved analytically whereas in Dynamic-Q, the solutions to the subproblems are approximated iteratively by the LfopC algorithm. This introduces a stochastic element to the solution procedure. It appears that this stochastic strategy is more efficient for the O synthesis problem. A further reason for the rejection of SQSD as the general algorithm of choice for the manipulator optimization problem, is that it is purely a method for unconstrained problems. This makes its use in constrained extensions to the manipulator optimization problem impossible.

All the algorithms tested are capable of solving the manipulator optimization problem given sufficient computing power. The function evaluations, each corresponding to a workspace computation, are however relatively expensive, and will become more so as manipulator dimensionality and complexity

increase. Efficiency is thus of primary importance. From the results obtained for the simple manipulator studied here it is expected that, for the more challenging manipulator design problems to be tackled later, Dynamic-Q will also be able to provide the efficiency required, with a minimal compromise on the accuracy of the solution.

4.6 Efficient inclusion of a prescribed workspace

The synthesis problem studied in this section is

> **E synthesis:** *determine the manipulator design resulting in a workspace which completely includes the prescribed workspace in the most efficient manner.*

4.6.1 Optimization formulation

One possible way of solving the E synthesis problem is by adjusting the weights w_c and w_p in (4.8). As w_c tends towards 0, δW_c takes on less importance in the optimization problem and the manipulator tends towards a design with a workspace that fully includes the prescribed workspace. The problem with this approach is that when $w_c = 0$, there are clearly an infinite number of solutions yielding the global optimum value $f^* = 0$, with as many of them corresponding to designs with unnecessarily large workspaces. This problem may be overcome by considering the following alternative *exact fit* constrained optimization problem:

$$\begin{aligned} &\min_{\mathbf{d}} f(\mathbf{d}) = \delta W_c \\ &\text{subject to the equality constraint} \\ &h(\mathbf{d}) = \delta W_p = 0 \end{aligned} \tag{4.14}$$

	N^g	f^*	h^*	$\mathbf{d}^*$
SP1	253	2.88511	$0.7E-2$	$[-4.152, -2.461, 4.060, 5.491, 5.441]^T$
SP2	573	2.85018	$0.3E-2$	$[-4.570, -2.203, 4.545, 5.542, 5.527]^T$
SP3	323	3.03526	$0.3E-2$	$[-4.023, -2.518, 4.235, 5.468, 5.563]^T$
SP4	402	3.15618	$0.1E-2$	$[-4.269, -2.567, 3.986, 5.618, 5.481]^T$

Table 4.7: E synthesis solutions

A solution to this problem corresponds to a manipulator design $\mathbf{d}^*$ with a corresponding workspace that fully encloses the prescribed workspace in an optimal manner, as required by the E synthesis statement.

4.6.2 Numerical results

The constrained optimization problem (4.14) was solved using the Dynamic-Q algorithm, with the same prescribed workspace (4.11) and starting points (Table 4.2 and Figure 4.4) as for the unconstrained problem. Termination parameters used for Dynamic-Q were $\varepsilon_f = 10^{-6}$ and $\varepsilon_x = 10^{-4}$. The results obtained for the different runs are summarized in Table 4.7 which gives the number of gradient evaluations (N^g), final objective f^* and equality constraint h^* function values, as well as the final design vector. Figure 4.10 shows the workspaces corresponding to the solutions obtained. For the results presented here, a chord length of $d = 0.1$ was used with number of boundary points $n_{bc} \approx 124$ at the solution. The finite difference interval for determining function gradients was $\Gamma = 10^{-3}$. Figure 4.11 gives the convergence histories from the four starting designs.

From Table 4.7 and Figure 4.10 it can be seen that the final workspaces obtained are grouped more closely together than for the unconstrained problem. This is to be expected with the introduction of the equality constraint. Of interest as well is that, although the resultant optimum designs and cor-

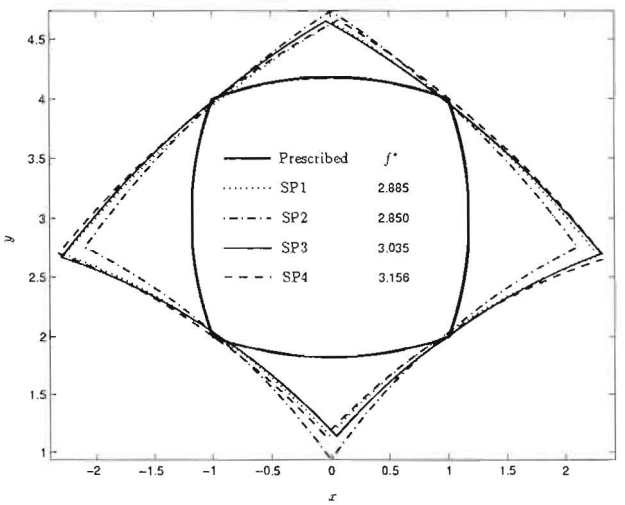

Figure 4.10: E synthesis workspaces corresponding to optimal designs

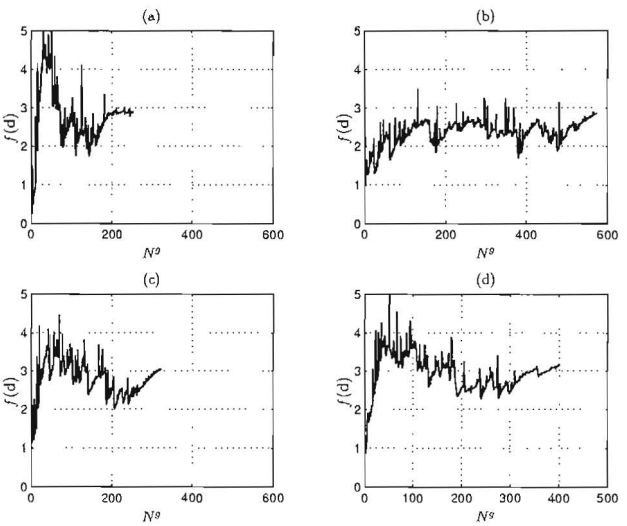

Figure 4.11: E synthesis convergence histories

responding workspaces for SP1 and SP2 are quite different from each other, their function values differ by less than 1%. This once more reinforces the previous conclusion that the objective functions considered here have relatively large flat regions near the optimum solution.

It is evident from inspection of Table 4.7 that the equality constraint h is not accurately satisfied at the converged solutions. This inaccuracy, and relatively slow and erratic convergence (see Figure 4.11) compared to that previously experienced (in Chapter 3), can be attributed to three possible factors. The first of these factors is the discrete nature of the workspace calculation, which may cause numerical noise in the problem. Indeed, as the chord length used in the workspace calculation is reduced, the equality constraint is more accurately satisfied, but at increased computational cost. The second factor influencing the poor convergence is the topography of the optimization problem near the solution. In particular, the choice that $l_i^{\max} = 1.5 l_i^{\min}, i = 1, 2$ seems to result in a flat-bedded, steep-sided valley near the solution, which in turn inhibits sharp convergence of the algorithm to the solution. A third possible reason for the erratic convergence behavior may be the obvious discontinuous nature of the gradient vector of $h(\mathbf{d})$ as design points at which exact fit is achieved are approached from design regions where $\delta W_p > 0$. This discontinuous nature cannot be modelled by the continuous quadratic approximations to $h(\mathbf{d})$ used by the Dynamic-Q algorithm.

4.7 Synthesis with respect to a performance measure

The final methodology proposed in this chapter is:

> **P synthesis:** *determine a manipulator design such that a prescribed workspace is fully enclosed by the manipulator workspace,*

and that the behavior is optimal, with respect to some performance measure, within the workspace.

4.7.1 Optimization formulation

Further investigation of the numerical ill-conditioning reported in Section 4.6.2, reveals that the dominant factor in introducing the poor and erratic convergence, is probably the the imposition of the leg-ratio equality constraint. It is concluded that this constraint is undesirable and in most design situations probably unnecessary. On the other hand it appears that the discrete manner in which the functions are computed does not seriously affect the conditioning of the problem. It is evident that for any prescribed workspace there are an infinite number of manipulator designs, even when restricted by some leg length ratio, which will result in a workspace that fully includes the prescribed one. Thus far, the criterion for choosing the most suitable design for the constrained cases was that the non-intersecting part of the calculated workspace area should be a minimum. As mentioned in the introduction, there are however a number of other factors, such as manipulator stiffness or conditioning of the workspace, which may be of even greater importance in parallel manipulator design. A possibly better approach to practical design may be, depending on the application, the maximization of, for example, the overall manipulator stiffness, subject to the constraint that the workspace of the optimal manipulator should include the prescribed workspace. In this respect the motivation for method presented here is similar to the Quaternion and Democrat methodologies discussed in Section 1.4.2. These methods determine the *set* of manipulator designs, the workspaces of which include prescribed points, or line segments. The most suitable manipulator with respect to some performance criterion or criteria can then be selected from this set. In this section a similarly motivated approach is developed and applied to the 2-dof planar parallel manipulator, although in this case the optimum manipulator is determined in one step by simultaneously

considering performance and workspace.

Thus the constrained optimization formulation, proposed here for the solution of the P synthesis problem, is aimed at optimizing manipulators with respect to some suitable performance index κ, which is to be reduced in an overall sense for the optimum performance of the manipulator, i.e.:

$$\begin{aligned} &\min_{\mathbf{d}} \left\{ \max_{\mathbf{u} \in W_p} \kappa(\mathbf{d}, \mathbf{u}) \right\} \\ &\text{subject to the inequality constraint} \\ &g(\mathbf{d}) \leq 0 \end{aligned} \tag{4.15}$$

where for a design $\mathbf{d}$, κ may be measured at any point $\mathbf{u}$ within the prescribed workspace W_p, and the inequality constraint function $g(\mathbf{d})$ is defined slightly differently to the equality constraint function $h(\mathbf{d})$ in (4.14). The displacement vector between the prescribed workspace boundary and calculated workspace boundary, measured along a ray emanating from $\mathbf{O}'$ at angle β_c^i is denoted by $r^i \mathbf{e}^i$, where $\mathbf{e}^i$ is a unit outward vector at angle β_c^i. If $r^{\min} = \min_i \{|r^i|, i = 1, 2, \ldots, n_{bc}\}$ set $r = r^{\min}$. The constraint function is now defined as follows:

$$g(\mathbf{d}) = \begin{cases} \delta W_p \text{ if } \delta W_p > 0 \\ -r^2 \text{ if } \delta W_p = 0 \end{cases} \tag{4.16}$$

This modification is made in order to improve the topography of the inequality constraint function by effectively avoiding the severe discontinuity in the gradient of $g(\mathbf{d})$, at the point of exact fit, that was previously present in the gradient of $h(\mathbf{d})$. It is believed that the severe discontinuities inherent in $h(\mathbf{d})$ adversely affected the performance of the Dynamic-Q algorithm (see last paragraph in Section 4.6.2).

The solution to optimization problem (4.15) seeks to improve the *single worst point* with respect to some chosen performance measure κ, within the *prescribed* workspace, W_p. This philosophy differs from that proposed by Gos-

selin and Angeles [74] and used by numerous other researchers where the *average* performance index over the *entire* workspace is optimized. Since it is assumed that the manipulator's movements will be limited to the prescribed workspace, it is only necessary to ensure good performance qualities within this workspace and thus it is better here to optimize the single worst value, instead of the average.

4.7.2 The condition number of the manipulator

The specific performance measure chosen here is the condition number of the Jacobian matrix of the manipulator, although any of the other criteria mentioned in Section 1.4.1 could also be used. The accuracy of control of the manipulator is dependent on the condition number (Gosselin and Angeles [74]). Since this quantity tends to infinity as the manipulator approaches a singular position, minimizing the condition number also ensures that the manipulator remains far away from such singular positions. For a general parallel manipulator the inverse kinematics are easy to solve.

From (4.3), an inverse transformation relating the input and output velocities can be determined:

$$\mathbf{J_u}\dot{\mathbf{u}} = -\mathbf{J_v}\dot{\mathbf{v}} \tag{4.17}$$

where $\mathbf{J_u}$ and $\mathbf{J_v}$ are the respective constraint Jacobian matrices, containing the partial derivatives of the m kinematic constraints (4.3) with respect to the variables $\mathbf{u}$ and $\mathbf{v}$. Equation (4.17) can be rewritten as

$$\mathbf{J}\dot{\mathbf{u}} = \dot{\mathbf{v}} \tag{4.18}$$

where $\mathbf{J} = -\mathbf{J_v^{-1}}\mathbf{J_u}$.

In general, the condition number κ of an $n \times n$ Jacobian $\mathbf{J}$ is defined as

$$\kappa = \|\mathbf{J}\|\|\mathbf{J}^{-1}\| \tag{4.19}$$

where $\|\cdot\|$ denotes any norm of its matrix argument. The norm adopted here is the same as that used by Gosselin and Angeles [74], namely

$$\|\mathbf{J}\| = \sqrt{\text{trace}(\mathbf{J}\mathbf{W}\mathbf{J}^{\mathsf{T}})} \tag{4.20}$$

where $\mathbf{W}$ is defined as n^{-1} multiplied by the $n \times n$ identity matrix. The lower the condition number, the better the behavior of the manipulator, with the lowest possible value of κ being unity. The value of κ^{-1}, the *inverse* of the condition number, thus lies between 0 and 1, and is preferably used in the objective function as it is bounded and better conditioned than the condition number itself. For the 2-dof parallel manipulator studied here $n = m = 2$ (see Section 4.3).

An optimization problem equivalent to (4.15) above is therefore:

$$\begin{aligned} &\max_{\mathbf{d}} \left\{ \min_{\mathbf{u} \in W_p} \kappa^{-1}(\mathbf{d}, \mathbf{u}) \right\} \\ &\text{subject to the inequality constraint} \\ &g(\mathbf{d}) \leq 0 \end{aligned} \tag{4.21}$$

where $g(\mathbf{d})$ is defined as in (4.16).

One point which arises concerns the nested part of optimization problem (4.21) and the question of how to determine the smallest value of κ^{-1} over the set $\mathbf{u} \in W_p$. Since we only require the single lowest value of the inverse condition number, an efficient method for determining this value, based on the necessary condition for an internal maximum or minimum of the condition number is proposed and used here.

The Jacobian $\mathbf{J}$ of the 2-dof manipulator ($nv = nu = 2,\ nw = 0$) shown in Figure 4.1 is given by

$$\mathbf{J} = \begin{bmatrix} (u_1 - x_A)/v_1 & (u_2 - y_A)/v_1 \\ (u_1 - x_B)/v_2 & (u_2 - y_B)/v_2 \end{bmatrix} \tag{4.22}$$

Using (4.19) and (4.20), and assuming $y_A = y_B$ the inverse condition number

κ^{-1} of the manipulator can be determined (see Appendix D):

$$\kappa^{-1}(\mathbf{u}) = \det(\mathbf{J}) = \frac{(u_2 - y_A)(x_B - x_A)}{v_1(\mathbf{u})v_2(\mathbf{u})} \tag{4.23}$$

The necessary condition for an internal maximum or minimum of the function $\kappa^{-1}(\mathbf{u})$ is that $\nabla\kappa^{-1}(\mathbf{u}) = \mathbf{0}$. It can be shown that the only set of solutions corresponding to this condition is

$$\mathbf{U} = \left\{\mathbf{u} \in \Re^{nu} \middle| \left[u_1 - (x_A + x_B)/2\right]^2 + \left[u_2 - y_A\right]^2 = \left[(x_A - x_B)/2\right]^2\right\} \tag{4.24}$$

which is the set of points on a circle of radius $(x_A - x_B)/2$ centered at $[(x_A+x_B)/2, y_A]^\top$. It can be shown that for these points $\kappa^{-1}(\mathbf{u}) = 1$, which is known to be the maximum possible value of κ^{-1}. The set $\mathbf{U}$ thus corresponds to a "maximum ridge" of the inverse condition number κ^{-1}. Now assume that there exists a point $\mathbf{u}' \in \Re^{nu}$ such that $\kappa^{-1}(\mathbf{u}')$ is a minimum. Since κ^{-1} is continuous for $y_P > y_A$ it is then necessary that $\nabla\kappa^{-1}(\mathbf{u}') = \mathbf{0}$. It has however been shown that the only solutions corresponding to the necessary conditions in the upper plane are elements of $\mathbf{U}$ and they are maxima. Thus the minimum value of κ^{-1} must lie on the boundary ∂W_p of the prescribed workspace W_p. A complete version of this proof is given in Appendix D.

The maximum value of the condition number can thus be approximated by calculating κ at points $\mathbf{b}_p^i$, $i = 1, \ldots, n_{bc}$ and then determining the overall maximum of the values at these candidate points. This of course also gives the corresponding overall minimum value of κ^{-1} required in (4.21).

4.7.3 Numerical results

Optimization problem (4.21) is solved for the 2-dof parallel manipulator with three different prescribed workspaces denoted P1-P3. Here it is assumed that the actuator sizes are fixed with $l_i^{\min} = 4.0, i = 1, 2$ and $l_i^{\max} = 7.0, i = 1, 2$. There are thus three design variables and the design vector is $\mathbf{d} = [x_A, y_A, x_B]^\top$. P1 is the workspace studied in the previous sections and is

$[\beta_{p\min}, \beta_{p\max})$	dx	dy	R	$\pm$
$[0, \pi/2)$	0.0000	0.0000	2.0000	+
$[\pi/2, \pi)$	-1.5000	0.0000	2.5000	+
$[\pi, 1.444\pi)$	2.7500	2.2500	2.8504	−
$[1.444\pi, 1.555\pi)$	-0.07005	0.86647	0.29206	−
$[1.555\pi, 2\pi)$	-2.3125	2.500	2.5195	−

Table 4.8: Parameters specifying prescribed workspace P3

described by (4.11). P2 is an ellipse centered at $\mathbf{O}' = [0, 3]^\top$ with x and y half axis lengths $a = 1.75$ and $b = 1.00$. P3 is a non-convex, non-symmetrical workspace centered at $\mathbf{O}' = [0, 3]^\top$ and defined in polar coordinates by

$$r_p(\beta_p) = -b \pm \sqrt{b^2 - dx^2 - dy^2 + R^2} \quad . \tag{4.25}$$
$$\text{where } b = dx \cos\beta_p + dy \sin\beta_p$$

and where the parameters dx, dy, R and the sign before the square root are, for various angular intervals $[\beta_{p\min}, \beta_{p\max})$, as given in Table 4.8. The boundary thus consists of five smooth arcs.

The three prescribed workspaces, as well as the workspace for the starting design $\mathbf{d}^0 = [-4, -0.1, 4]^\top$ are shown in Figure 4.12(a). Contours of the reciprocal of the condition number κ^{-1} corresponding to the starting design are also plotted. Optimization problem (4.21) was implemented using the Dynamic-Q algorithm with move limit $\rho = 0.2$ and chord length of $d = 0.1$ for calculating the workspace. Termination criteria $\varepsilon_f = 10^{-4}$ and $\varepsilon_x = 10^{-3}$ were used. For each prescribed workspace P1-P3, Table 4.9 summarizes the number of gradient evaluations N^g required to reach the optimum solution f^* from the starting function value $f(\mathbf{d}^0)$, as well as the value of the inequality constraint function g^* at the solution and the solution vector $\mathbf{d}^*$. Workspaces corresponding to the solutions along with plots of the reciprocal of the condition number are given in Figures 4.12(b) and 4.13. Convergence histories corresponding to Table 4.9 are given in Figure 4.14.

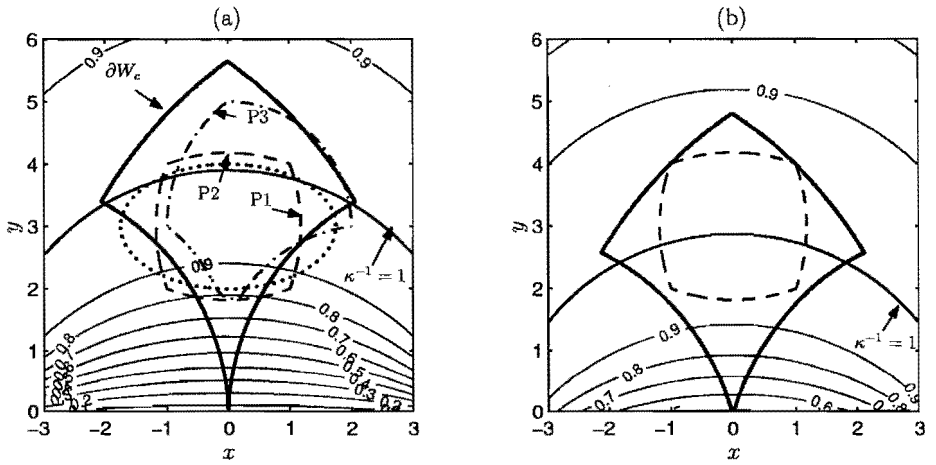

Figure 4.12: P synthesis (a) prescribed workspaces P1-P3, manipulator workspace and κ^{-1} contours corresponding to the starting design and (b) prescribed workspace P1 and corresponding optimal manipulator workspace and κ^{-1} contours

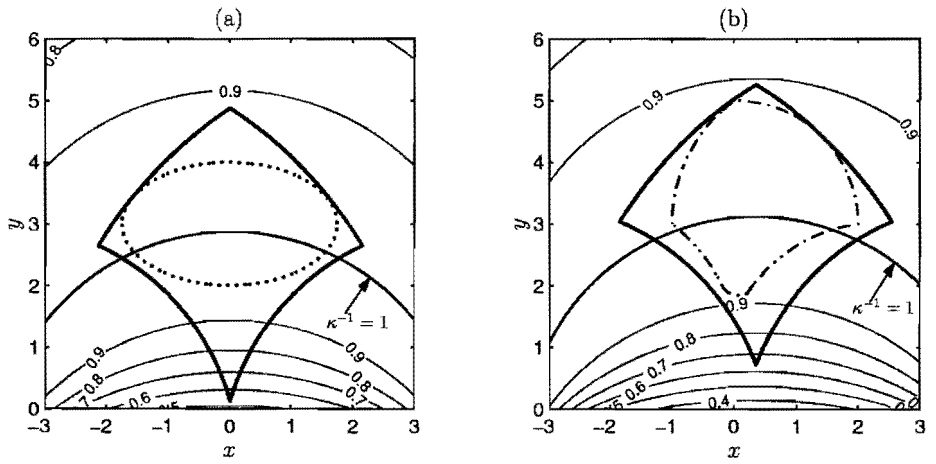

Figure 4.13: P synthesis manipulator workspace and κ^{-1} contours corresponding to the optimal design for prescribed workspaces (a) P2 and (b) P3

	N^g	$f(\mathbf{d}_0)$	f^*	g^*	$\mathbf{d}^*$
P1	10	0.780	0.952	$-0.5E-5$	$[-3.892, -1.016, 3.882]^\mathsf{T}$
P2	12	0.823	0.968	$-0.2E-5$	$[-3.830, -0.9740, 3.859]^\mathsf{T}$
P3	10	0.786	0.921	$-0.5E-4$	$[-3.414, -0.6427, 4.107]^\mathsf{T}$

Table 4.9: P synthesis solutions

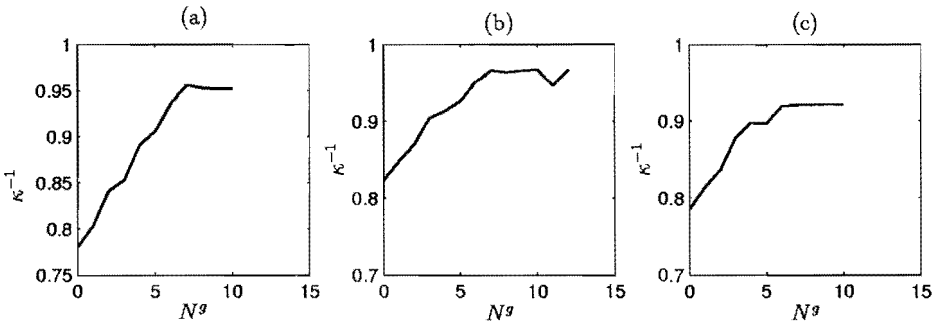

Figure 4.14: P synthesis convergence histories for (a) P1, (b) P2, and (c) P3

The results obtained are extremely encouraging, accurate and optimal solutions having been obtained with minimal computational effort. In each case the algorithm not only determines manipulator dimensions so that the prescribed workspace can be reached by the manipulator, but also places the calculated workspace so that the condition number is as low as possible throughout the prescribed workspace.

4.8 Conclusion

Various computational schemes for synthesizing planar parallel manipulator designs have been proposed, successfully implemented and evaluated. In broad terms, two different criteria are applied in measuring the correspondence. The application of the first criterion is equivalent to seeking a good overall approximation of the prescribed workspace and results in an

unconstrained optimization problem. The second criterion requires that the prescribed workspace be fully contained in the optimal manipulator workspace and results in a constrained optimization problem. The unconstrained formulation solving the O synthesis problem has been used to select the most suitable algorithm for this class of problems from four different algorithms developed at the University of Pretoria. The Dynamic-Q method exhibits higher efficiency than the other methods under consideration. In the search for an efficient, stable, well-conditioned and practically relevant method two different formulations of the constrained optimization problem, namely E and P synthesis formulations, were proposed and numerically evaluated. Although the Dynamic-Q algorithm was successfully applied to the constrained optimization problems, its performance, with respect to accuracy and convergence rate, appeared to be seriously affected by the apparent poor conditioning of the E synthesis formulation. This was initially ascribed to both the inherent discrete manner in which the objective and constraint functions are computed and to the topographical ill-conditioning introduced by the imposition of the leg ratio equality constraint. A third likely reason for the erratic convergence behavior is thought to be the discontinuous nature of the prescribed equality constraint. Nevertheless, from an engineering design point of view, good solutions are obtained. Finally, the P synthesis formulation, which attempts to obtain a well-conditioned manipulator workspace which fully contains the prescribed workspace, was proposed. In this formulation the severe discontinuity previously present in the constraint function, which adversely influenced the performance of Dynamic-Q, was effectively removed. This final methodology produces convincing results giving a stable and efficient method for designing 2-dof planar parallel manipulators. Although the search for a fundamentally sound and robust numerical methodology for synthesizing parallel manipulators was restricted to the 2-dof planar case, it nevertheless led to a successful methodology that appears to be general. It seems that that the final P synthesis methodology is possibly the most practically relevant one.