Appendix A

Test functions used for SQSD

Minimize $f(\mathbf{x})$:

1. $f(\mathbf{x}) = x_1^2 + 2x_2^2 + 3x_3^2 - 2x_1 - 4x_2 - 6x_3 + 6,\ \mathbf{x}^0 = [3,3,3]^\top,\ \mathbf{x}^* = [1,1,1]^\top,\ f(\mathbf{x}^*) = 0.0$
2. $f(\mathbf{x}) = x_1^4 - 2x_1^2x_2 + x_1^2 + x_2^2 - 2x_1 + 1,\ \mathbf{x}^0 = [3,3]^\top,\ \mathbf{x}^* = [1,1]^\top,\ f(\mathbf{x}^*) = 0.0$
3. $f(\mathbf{x}) = x_1^4 - 8x_1^3 + 25x_1^2 + 4x_2^2 - 4x_1x_2 - 32x_1 + 16,\ \mathbf{x}^0 = [3,3]^\top,\ \mathbf{x}^* = [2,1]^\top,\ f(\mathbf{x}^*) = 0.0$
4. $f(\mathbf{x}) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2,\ \mathbf{x}^0 = [-1.2,1]^\top,\ \mathbf{x}^* = [1,1]^\top,\ f(\mathbf{x}^*) = 0.0$ (Rosenbrock's parabolic valley [113])
5. $f(\mathbf{x}) = x_1^4 + x_1^3 - x_1 + x_2^4 - x_2^2 + x_2 + x_3^2 - x_3 + x_1x_2x_3$, (Zlobec's function [114])
   (a) $\mathbf{x}^0 = [1,-1,1]^\top$ and
   (b) $\mathbf{x}^0 = [0,0,0]^\top,\ \mathbf{x}^* = [0.57085597, -0.93955591, 0.76817555]^\top,\ f(\mathbf{x}^*) = -1.91177218907$

6. $f(\mathbf{x}) = (x_1 + 10x_2)^2 + 5(x_3 - x_4)^2 + (x_2 - 2x_3)^4 + 10(x_1 - x_4)^4$, $\mathbf{x}^0 = [3, -1, 0, 1]^\top$, $\mathbf{x}^* = [0, 0, 0, 0]^\top$, $f(\mathbf{x}^*) = 0.0$ (Powell's quartic function [113])

7. $f(\mathbf{x}) = -\left\{\frac{1}{1+(x_1-x_2)^2} + \sin\left(\frac{1}{2}\pi x_2 x_3\right) + \exp\left[-\left(\frac{x_1+x_3}{x_2} - 2\right)^2\right]\right\}$, $\mathbf{x}^0 = [0, 1, 2]^\top, \mathbf{x}^* = [1, 1, 1]^\top$, $f(\mathbf{x}) = -3.0$ [113]

8. $f(\mathbf{x}) = \{-13 + x_1 + [(5 - x_2)x_2 - 2]x_2\}^2 + \{-29 + x_1 + [(x_2 + 1)x_2 - 14]x_2\}^2$, $\mathbf{x}^0 = [1/2, -2]^\top$, $\mathbf{x}^* = [5, 4]^\top, f(\mathbf{x}^*) = 0.0$ (Freudenstein and Roth function [113])

9. $f(\mathbf{x}) = 100(x_2 - x_1^3)^2 + (1 - x_1)^2$, $\mathbf{x}^0 = [-1.2, 1]^\top$, $\mathbf{x}^* = [1, 1]^\top$, $f(\mathbf{x}^*) = 0.0$ (cubic valley [115])

10. $f(\mathbf{x}) = [1.5 - x_1(1 - x_2)]^2 + [2.25 - x_1(1 - x_2^2)]^2 + [2.625 - x_1(1 - x_2^3)]^2$, $\mathbf{x}^0 = [1, 1]^\top$, $\mathbf{x}^* = [3, 1/2]^\top$, $f(\mathbf{x}^*) = 0.0$ (Beale's function [113])

11. $f(\mathbf{x}) = [10(x_2 - x_1^2)]^2 + (1 - x_1)^2 + 90(x_4 - x_3^2)^2 + (1 - x_3)^2 + 10(x_2 + x_4 - 2)^2 + 0.1(x_2 - x_4)^2$, $\mathbf{x}^0 = [-3, 1, -3, -1]^\top$, $\mathbf{x}^* = [1, 1, 1, 1]^\top$, $f(\mathbf{x}^*) = 0.0$ (Wood's function [113])

12. $f(\mathbf{x}) = \sum_{i=1}^{n} i x_i^2$, $\mathbf{x}^0 = [3, 3, \ldots, 3]^\top$, $\mathbf{x}^* = [0, 0, \ldots, 0]^\top$, $f(\mathbf{x}^*) = 0.0$ (extended homogeneous quadratic functions)

13. $f(\mathbf{x}) = \sum_{i=1}^{n-1}[100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2]$, $\mathbf{x}^0 = [-1.2, 1, -1.2, 1, \ldots]^\top$, $\mathbf{x}^* = [1, 1, \ldots, 1]^\top$, $f(\mathbf{x}^*) = 0.0$ (extended Rosenbrock functions [113])

14. $f(\mathbf{x}) = \sum_{i=1}^{n}(1 - x_i)^2/2^{i-1}$, $\mathbf{x}^0 = [0, 0, \ldots, 0]^\top$, $\mathbf{x}^* = [1, 1, \ldots, 1]^\top$, $f(\mathbf{x}^*) = 0.0$ (extended Manevich functions [116])

Appendix B

The dynamic trajectory optimization algorithm

B.1 Background

The dynamic trajectory method (also called the "leap-frog" method) for the *unconstrained* minimization of a scalar function $f(\mathbf{x})$ of n real variables represented by the vector $\mathbf{x} = [x_1, x_2, \ldots, x_n]^T$ was originally proposed by Snyman [101, 102]. The original algorithm has also been modified to handle constrained problems by means of a penalty function formulation (Snyman *et al.* [119] and Snyman [103]). The method possesses the following characteristics:

- It uses only function *gradient* information $\nabla f(\mathbf{x})$,
- *No* explicit *line searches* are performed,
- It is extremely *robust*: handles steep valleys and discontinuities in functions and gradients,
- Algorithm seeks a *low local minimum* and can therefore be used as a

basic component in a methodology for global minimization,

- The method is not as efficient as classical methods on smooth or near-quadratic functions

B.2 Basic dynamic model

In its unconstrained form, the leap-frog optimizer (Lfop) determines the minimum of a function $f(\mathbf{x})$, by considering the associated dynamic problem of the motion of a particle (of unit mass) in an n-dimensional conservative force field, where the potential energy of the particle at a point $\mathbf{x}(t)$ and time t, is given by $f(\mathbf{x})$. The method thus requires the solution of the equations of motion of the particle. At $\mathbf{x}$ the force on the particle is given by

$$\mathbf{a} = \ddot{\mathbf{x}}(t) = -\nabla f(\mathbf{x}(t)) \tag{B.1}$$

subject to initial conditions

$$\mathbf{x}(0) = \mathbf{x}^0, \ \dot{\mathbf{x}}(0) = \mathbf{v}^0 \tag{B.2}$$

To explain how the dynamic trajectory method works, consider the solution of the above problem over the time interval $[0, t]$. It follows that

$$\begin{aligned} \tfrac{1}{2}\|\dot{\mathbf{x}}(t)\|^2 - \tfrac{1}{2}\|\mathbf{v}^0\|^2 &= f(\mathbf{x}^0) - f(\mathbf{x}(t)) \\ T(t) - T(0) &= f(0) - f(t) \\ \text{or } f(t) + T(t) &= f(0) + T(0) = K \end{aligned} \tag{B.3}$$

Here $T(t)$ is used to denote the kinetic energy of the particle at time t and K is a constant determined by the initial values. The last expression in (B.3) indicates that energy is conserved. It can also be seen that $\Delta f = -\Delta T$, therefore as long as T increases, f decreases. This forms the basis of the dynamic trajectory method.

B.3 Basic algorithm for unconstrained problems

Given $f(\mathbf{x})$, and starting point $\mathbf{x}(0) = \mathbf{x}^0$, the Lfop algorithm computes an approximation to the trajectory followed by the particle in the force field by solving the initial value problem (B.1) and (B.2).

The algorithm monitors the velocity $\dot{\mathbf{x}}(t) = \mathbf{v}(t)$ of the particle. Clearly as long as $T = \frac{1}{2}\|\mathbf{v}(t)\|^2$ is *increasing* along the trajectory, $f(\mathbf{x}(t))$ is decreasing and the algorithm is minimizing the function. However, whenever T *decreases* along the trajectory, the objective function (potential energy) is increasing. An interfering strategy is then applied to extract kinetic energy from the particle. The consequence of this strategy, based on an energy conservation argument, is that a systematic reduction in the potential energy $f(\mathbf{x})$ of the particle is obtained, and the likelihood of descent is increased. The particle is thus forced to follow a path to a local minimum at $\mathbf{x}^*$.

The numerical integration of the initial value problem (B.1) and (B.2) is achieved using the "leap-frog" (Euler forward-Euler backward) method, by computing for $k = 0, 1, 2, \ldots$, and time step Δt

$$\begin{aligned} \mathbf{x}^{k+1} &= \mathbf{x}^k + \mathbf{v}^k \Delta t \\ \mathbf{v}^{k+1} &= \mathbf{v}^k + \mathbf{a}^{k+1} \Delta t \end{aligned} \tag{B.4}$$

where $\mathbf{a}^{k+1} = -\nabla f(\mathbf{x}^{k+1})$ and $\mathbf{v}^0 = \frac{1}{2}\mathbf{a}^0 \Delta t$.

A typical interfering strategy, implemented when $\|\mathbf{v}^{k+1}\| \leq \|\mathbf{v}^k\|$ is to set

$$\mathbf{v}^k = \frac{\mathbf{v}^{k+1} + \mathbf{v}^k}{4} \text{ and } \mathbf{x}^k = \frac{\mathbf{x}^{k+1} + \mathbf{x}^k}{2} \tag{B.5}$$

and then use these new values of $\mathbf{v}^k$ and $\mathbf{x}^k$ to compute the new $\mathbf{v}^{k+1}$ using (B.4).

The method contains some heuristic elements to determine an initial Δt, to

allow for reduction and magnification of Δt, and to control the step size used in the algorithm.

B.4 Modification for constrained problems

The Lfop algorithm outlined above can be modified to handle constrained problems by means of the penalty function approach (LfopC) [119, 103]. Given a *constrained* optimization problem with objective function $f(\mathbf{x})$, $\mathbf{x} \in \Re^n$, inequality constraint functions $g_j(\mathbf{x}) \leq 0$, $j = 1, 2, \ldots, p$, equality constraint functions $h_k(\mathbf{x}) \leq 0$, $k = 1, 2, \ldots, q$ and penalty parameters α_j and β_k, the associated *unconstrained* optimization problem given by the penalty function formulation is

$$Q(\mathbf{x}) = f(\mathbf{x}) + \sum_{j=1}^{p} \varrho_j g_j^2(\mathbf{x}) + \sum_{k=1}^{q} \beta_k h_k^2(\mathbf{x}) \tag{B.6}$$

$$\text{where } \varrho_j = \begin{cases} 0 \text{ if } g_j(\mathbf{x}) \leq 0 \\ \alpha_j \text{ if } g_j(\mathbf{x}) > 0 \end{cases}$$

For simplicity the penalty parameters α_j and β_k usually take on the same positive value $\alpha_j = \beta_k = \mu$. It can be shown that as μ tends to infinity, the unconstrained minimum of $Q(\mathbf{x})$ yields the solution to the constrained optimization problem. The Lfop dynamic trajectory method is applied to the penalty function formulation of the constrained problem in three phases.

PHASE 0: Given some starting point $\mathbf{x}^0$, apply Lfop with some overall penalty parameter $\mu = \mu_0 (= 10^2)$ to $Q(\mathbf{x}, \mu_0)$ to give $\mathbf{x}^*(\mu_0)$.

PHASE 1: With $\mathbf{x}^0 := \mathbf{x}^*(\mu_0)$, apply Lfop with increased overall penalty parameter $\mu = \mu_1 (= 10^4) \gg \mu_0$ to $Q(\mathbf{x}, \mu_1)$ to give $\mathbf{x}^*(\mu_1)$. Identify the set of n_a active constraints corresponding to the set of subscripts $I_a = (u1, u2, \ldots, un_a)$ for which $g_{uj}(\mathbf{x}^*(\mu_1)) > 0$, $j = 1, 2, \ldots, n_a$.

PHASE 2: With $\mathbf{x}^0 := \mathbf{x}^*(\mu_1)$, apply Lfop to

$$\min_{\mathbf{x}} Q_a(\mathbf{x}, \mu_1) = \sum_{j=1}^{n_a} \mu_1 g_{uj}^2(\mathbf{x}) + \sum_{k=1}^{q} \mu_1 h_k^2(\mathbf{x}) \tag{B.7}$$

to give $\mathbf{x}^*$.

Appendix C

Review of the chord method for workspace determination

C.1 Introduction

This appendix summarizes the chord method for maximal workspace determination as proposed by Hay and Snyman [50]. Further references related to the chord method, and the optimization approach for workspace determination are given in Section 1.3.2. In the next two sections of this appendix, definitions necessary for the discussion of the workspace determination method are given. The most important basic components of the chord method are determining an initial point on the workspace boundary, mapping the workspace boundary using constant chord length searches, and the accurate determination of bifurcation points which occur along the workspace boundary. Each of these components of are discussed in separate sections.

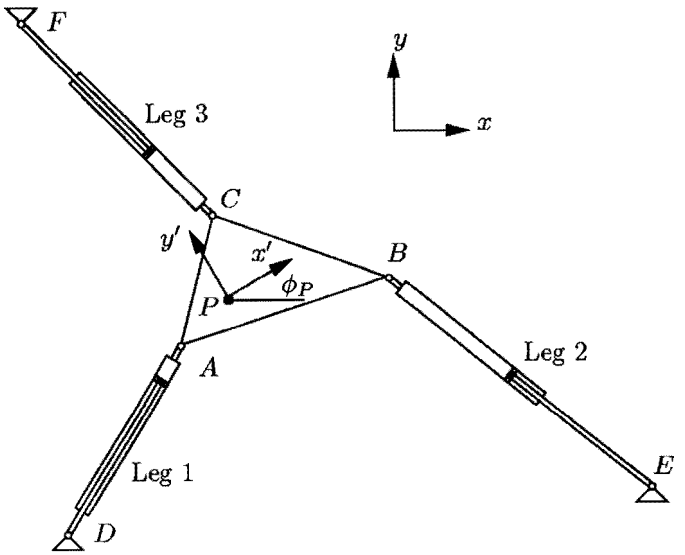

Figure C.1: A general planar parallel manipulator

C.2 A general planar parallel manipulator

Consider the planar parallel manipulator shown in Figure C.1, for which the input coordinates are the leg lengths $\mathbf{v} = [l_1, l_2, l_3]^\top$, the output coordinates are the coordinates of the working point P of the moving platform $\mathbf{u} = [x_P, y_P]^\top$, and the remaining intermediate coordinate is the orientation of the moving platform $w = \phi_P$. Refer to Section 4.2 for definitions of these coordinates.

Physical limits on the input variables (leg lengths) take the form of inequality constraints

$$\mathbf{v}^{\min} \leq \mathbf{v} \leq \mathbf{v}^{\max} \tag{C.1}$$

Similarly, any other physical limits which might be imposed by the manipulator construction (such as limits on the passive joint angles) can be expressed in terms of input, output and intermediate variables by

$$\mathbf{g}^{\min} \leq \mathbf{g}(\mathbf{u}, \mathbf{v}, w) \leq \mathbf{g}^{\max} \tag{C.2}$$

The vectors $\mathbf{v}^{\min}$, $\mathbf{v}^{\max}$, $\mathbf{g}^{\min}$ and $\mathbf{g}^{\max}$ contain the numerical lower and upper

limits to the constraints specified in (C.1) and (C.2).

C.3 Maximal workspace definition

The kinematic constraint equations (4.1), ensuring assembly of the mechanism, may be rewritten in terms of the defined coordinates:

$$\mathbf{\Phi}(\mathbf{u}, \mathbf{v}, w) = \mathbf{0} \tag{C.3}$$

The *maximal* workspace W^M of the manipulator, in agreement with the definition given in Section 1.3.1, is defined as

$$\begin{aligned} W^M \;=\; & \{\mathbf{u} \in \Re^{nu} : \mathbf{\Phi}(\mathbf{u}, \mathbf{v}, w) = \mathbf{0};\ \mathbf{v} \text{ satisfying (C.1)}; \\ & \mathbf{g}(\mathbf{u}, \mathbf{v}, w) \text{ satisfying (C.2)}\} \end{aligned} \tag{C.4}$$

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

$$\begin{aligned} \partial W^M \;=\; & \{\mathbf{u} \in \Re^{nu} : \mathbf{u} \in W^M \text{ and } \exists \text{ an } \mathbf{s} \in \Re^{nu} \text{ such that for} \\ & \mathbf{u}' = \mathbf{u} + \lambda \mathbf{s},\ \lambda \in \Re \text{ arbitrarily small and either} \\ & \text{positive or negative, no } \mathbf{v} \text{ and } w \text{ exist that satisfy} \\ & \mathbf{\Phi}(\mathbf{u}', \mathbf{v}, w) = \mathbf{0} \text{ as well as inequalities (C.1) – (C.2)}\} \end{aligned} \tag{C.5}$$

C.4 Finding an initial point of the workspace boundary

As a starting point for the chord method, a suitable radiating point $\mathbf{u}^0$ must be found within the workspace. For a parallel manipulator, the inverse kinematics are easy to solve. Thus, given $\mathbf{u}$ and w, system (C.3) may easily be solved to give $\mathbf{v}$ in terms of $\mathbf{u}$ and w:

$$\mathbf{v} = \mathbf{v}(\mathbf{u}, w) \tag{C.6}$$

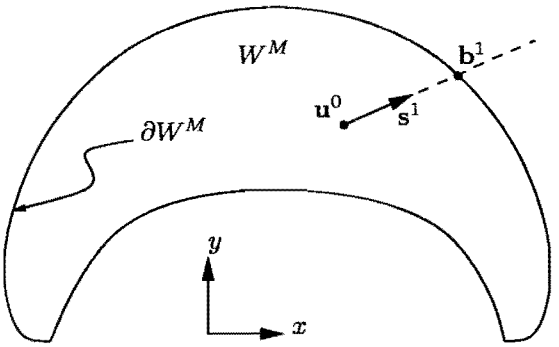

Figure C.2: Finding an initial point on the maximal workspace boundary

Since we are analyzing a planar manipulator, the maximal workspace W^M is two-dimensional. Depending on the particular geometry of the manipulator a suitable choice for a radiating point $\mathbf{u}^0$, inside the workspace, may be self-evident. If not, then $\mathbf{u}^0$ may be obtained from (C.6) by solving for $\mathbf{u}$ and w in:

$$\bar{\mathbf{v}} = \mathbf{v}(\mathbf{u}, w) \tag{C.7}$$

where $\bar{\mathbf{v}} = (\mathbf{v}^{\min} + \mathbf{v}^{\max})/2$.

In practice this can be done by solving the least squares optimization problem

$$\min_{\mathbf{u},w} |\mathbf{v}(\mathbf{u}, w) - \bar{\mathbf{v}}\|^2 \tag{C.8}$$

Consistent with the definition of W^M in (C.5), an initial point $\mathbf{b}^1 = (x^1, y^1)^\top$ on the boundary, in an arbitrarily chosen direction (designated by a unit vector $\mathbf{s}^1$) from $\mathbf{u}^0$, may be determined by solving the following constrained optimization problem:

$$\begin{gathered} \max_{\mathbf{u},w} \|\mathbf{u} - \mathbf{u}^0\| \\ \text{subject to } \mathbf{v}^{\min} \leq \mathbf{v}(\mathbf{u}, w) \leq \mathbf{v}^{\max}; \\ \mathbf{g}^{\min} \leq \mathbf{g}(\mathbf{u}, \mathbf{v}, w) \leq \mathbf{g}^{\max} \\ \text{and equality constraint } h(\mathbf{u}, \mathbf{s}^1) = \frac{\mathbf{u}-\mathbf{u}^0}{\|\mathbf{u}-\mathbf{u}^0\|} \cdot \mathbf{s}^1 - 1 = 0 \end{gathered} \tag{C.9}$$

The solution of Problem (C.9) is illustrated in Figure C.2. Essentially optimization problem (C.9) seeks to find the point of intersection ($\mathbf{b}^1$) of a ray,

emanating from $\mathbf{u}^0$ in direction $\mathbf{s}^1$, with the workspace boundary.

C.5 Basic chord methodology

Consider any boundary point $\mathbf{b}^i$ with an associated *unit* vector $\mathbf{s}^{1i}$ pointing *out* of the workspace. A vector $\mathbf{s}^{2i}$ from $\mathbf{b}^i$ to an arbitrary output point $\mathbf{u}$, corresponding to a position of the working point of the manipulator, is

$$\mathbf{s}^{2i} = \mathbf{u} - \mathbf{b}^i \tag{C.10}$$

Dropping the superscript i, the angle ω between the unit vector $\mathbf{s}^1$ and vector $\mathbf{s}^2$, defined in the right hand sense, is given by

$$\omega = \begin{cases} \cos^{-1}\left(\frac{\mathbf{s}^1 \cdot \mathbf{s}^2}{\|\mathbf{s}^2\|}\right) & \text{if } \alpha \geq 0 \\ 2\pi - \cos^{-1}\left(\frac{\mathbf{s}^1 \cdot \mathbf{s}^2}{\|\mathbf{s}^2\|}\right) & \text{if } \alpha < 0 \end{cases} \tag{C.11}$$

where $\mathbf{s}^1 \times \mathbf{s}^2 = \alpha\hat{z}$ and $\hat{z}$ is the unit vector in the $z-$direction.

Clearly ω is a function of the output coordinates $\mathbf{u}$.

Given such a point $\mathbf{b}^i$ on the workspace boundary, the next point at a constant chord length d along the workspace boundary may be determined by means of a modified version of optimization problem (C.9):

$$\begin{gathered} \min_{\mathbf{u},w} \omega \\ \text{subject to } \mathbf{v}^{\min} \leq \mathbf{v}(\mathbf{u}, w) \leq \mathbf{v}^{\max}; \\ \mathbf{g}^{\min} \leq \mathbf{g}(\mathbf{u}, \mathbf{v}, w) \leq \mathbf{g}^{\max} \\ \text{and equality constraint } h = \|\mathbf{u} - \mathbf{b}^i\| - d = 0 \end{gathered} \tag{C.12}$$

The solution to this optimization problem is depicted in Figure C.3. Having solved problem (C.12), $\mathbf{s}^{2i} = \mathbf{b}^{i+1} - \mathbf{b}^i$ with components s_x^{2i} and s_y^{2i} is precisely known and a *new* reference vector $\mathbf{s}^{2(i+1)}$, associated with the new boundary point $\mathbf{b}^{i+1}$ can now be determined as follows:[1]

[1]The function $\tan 2^{-1}$ has two input arguments and returns an answer in the range $[0, 2\pi]$.

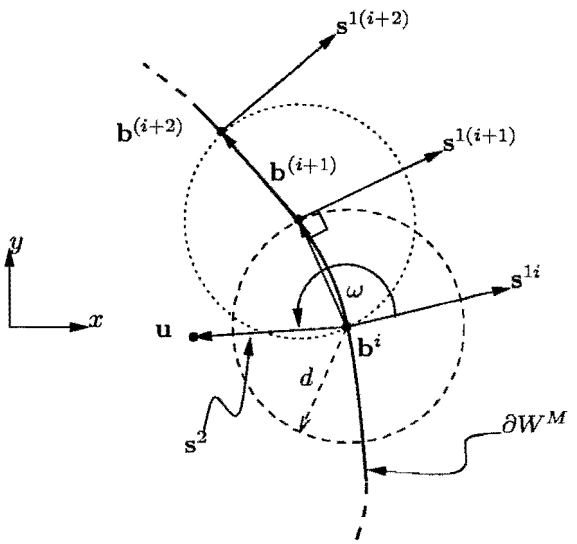

Figure C.3: The chord methodology

$$\mathbf{s}^{1(i+1)} = \begin{bmatrix} \cos\left(\tan 2^{-1}\left(\frac{s_y^{2i}}{s_x^{2i}}\right) - \frac{\pi}{2}\right) \\ \sin\left(\tan 2^{-1}\left(\frac{s_y^{2i}}{s_x^{2i}}\right) - \frac{\pi}{2}\right) \end{bmatrix} \tag{C.13}$$

This equation defines a vector perpendicular to $\mathbf{s}^{2i}$ and pointing out of the workspace. Since it has already been shown how an initial point $\mathbf{b}^1$ and reference vector can be found, it follows that the workspace boundary ∂W^M can be mapped numerically by successively solving optimization problem (C.12) for $i = 1, 2, 3, \ldots$, each time using the solution to the previous problem as the starting point for the new optimization problem. Equation (C.13) is used to determine the associated reference vector for each new boundary point, $\mathbf{b}^{i+1}$.

The algorithm is terminated when a specified maximum number of iterations is exceeded or when

$$\begin{aligned} \|\mathbf{b}^i - \mathbf{b}^1\| &\leq d \\ \text{and } \|\mathbf{b}^i - \mathbf{b}^2\| &\leq d \end{aligned} \tag{C.14}$$

which is an indication of closure of the workspace boundary.

The formulation given above maps the workspace boundary in a counter-clockwise manner. In order to map in the clockwise direction it is necessary to modify the definition of ω in (C.11) to

$$\omega = \begin{cases} \cos^{-1}\left(\frac{\mathbf{s}^1 \cdot \mathbf{s}^2}{\|\mathbf{s}^2\|}\right) & \text{if } \alpha \leq 0 \\ 2\pi - \cos^{-1}\left(\frac{\mathbf{s}^1 \cdot \mathbf{s}^2}{\|\mathbf{s}^2\|}\right) & \text{if } \alpha > 0 \end{cases} \qquad \text{where } \mathbf{s}^1 \times \mathbf{s}^2 = \alpha \hat{z} \tag{C.15}$$

and to change calculation (C.13) of the reference vector $\mathbf{s}^{1(i+1)}$ to

$$\mathbf{s}^{1(i+1)} = \begin{bmatrix} \cos\left(\tan 2^{-1}\left(\frac{s_y^{2i}}{s_x^{2i}}\right) + \frac{\pi}{2}\right) \\ \sin\left(\tan 2^{-1}\left(\frac{s_y^{2i}}{s_x^{2i}}\right) + \frac{\pi}{2}\right) \end{bmatrix} \tag{C.16}$$

The same termination conditions (C.14) apply.

C.6 Determination of bifurcation points

Whenever the manipulator moves along a trajectory such that motion is restricted in some direction, the manipulator is said to be moving along a *bifurcation* path. For maximal workspaces of the planar 3-dof manipulators, such paths correspond to configurations either when two legs remain at extreme lengths while the third varies between extreme values or when one leg is at an extreme length and remains collinear with the working point while the others vary between extreme values. As is to be expected, the *boundary* of the maximal workspace consists of portions of bifurcation paths. The remaining portions of the bifurcation paths are internal bifurcation paths, which correspond to positions within the workspace where the manipulator motion is restricted (Haug *et al.* [32]). Points of intersection of bifurcation paths in the output space usually correspond to bifurcation *points*.

Bifurcation points are positions in the output space where the manipulator can branch or change directly from following one bifurcation path to another.

It is necessary to determine the exact positions of bifurcation points to ensure accurate mapping of the workspace boundary. The precise determination of bifurcation points is done by, having identified the constraints active at the bifurcation point during the mapping procedure, then solving for the point of intersection of these constraints using an optimization approach. Further details of this approach can be found in Snyman *et al.* [48], and Hay and Snyman [49, 50].

Appendix D

Location of the minimum value of κ^{-1} for the two-dof manipulator

THEOREM. *For the 2-dof planar parallel manipulator with $y_A = y_B$, $x_B > x_A$ and $y > y_A$, and any arbitrarily chosen prescribed workspace W_p, the minimum value of κ^{-1} always lies on the boundary ∂W_p of the workspace W_p.*

PROOF. Consider the 2-dof planar parallel manipulator shown in Figure 4.1. The coordinates of point P (output coordinates) are $\mathbf{u} = [x_P, y_P]^\top = [u_1, u_2]^\top$, the input coordinates are $\mathbf{v} = [l_1, l_2]^\top = [v_1, v_2]^\top$. The generalized coordinates are thus $\mathbf{q} = [\mathbf{u}, \mathbf{v}]^\top = [x_P, y_P, l_1, l_2]^\top$.

Dropping the subscript P when referring to x_P and y_P, the actuator leg lengths are given by

$$\begin{aligned} l_1^2 &= (x - x_A)^2 + (y - y_A)^2 \\ l_2^2 &= (x - x_B)^2 + (y - y_B)^2 \end{aligned} \tag{D.1}$$

The kinematic constraints $\mathbf{\Phi(q)}$ are thus

$$\mathbf{\Phi(q)} = \begin{bmatrix} l_1^2 - (x - x_A)^2 - (y - y_A)^2 \\ l_2^2 - (x - x_B)^2 - (y - y_B)^2 \end{bmatrix} = \mathbf{0} \tag{D.2}$$

Differentiation of (D.2) with respect to time yields

$$\mathbf{\Phi_q \dot{q}} = \mathbf{0} \tag{D.3}$$

where $\mathbf{\Phi_q}$ contains the partial derivatives of (D.2) with respect to $\mathbf{q}$.

$$\mathbf{\Phi_q} = \left[\begin{array}{cc|cc} -2(x - x_A) & -2(y - y_A) & 2l_1 & 0 \\ -2(x - x_B) & -2(y - y_B) & 0 & 2l_2 \end{array}\right] = [\mathbf{J_u}|\mathbf{J_v}] \tag{D.4}$$

Here the partitioning separates $\mathbf{\Phi_q}$ into entries related to the input coordinates, denoted $\mathbf{J_u}$ and entries related the output coordinates, denoted $\mathbf{J_v}$. Using this partitioning, (D.3) can be manipulated as follows:

$$\begin{aligned} \mathbf{\Phi_q \dot{q}} = [\mathbf{J_u}|\mathbf{J_v}] \begin{bmatrix} \dot{\mathbf{u}} \\ \hline \dot{\mathbf{v}} \end{bmatrix} &= \mathbf{J_u \dot{u}} + \mathbf{J_v \dot{v}} = \mathbf{0} \\ \mathbf{J_u \dot{u}} &= -\mathbf{J_v \dot{v}} \\ \mathbf{\dot{v}} &= -\mathbf{J_v}^{-1}\mathbf{J_u \dot{u}} = \mathbf{J \dot{u}} \end{aligned} \tag{D.5}$$

where $\mathbf{J}$ is the Jacobian of the manipulator. Substituting the expressions for $\mathbf{J_u}$ and $\mathbf{J_v}$, contained in (D.4), into (D.5), an explicit expression for the Jacobian of the 2-dof planar manipulator may be obtained.

$$\begin{aligned} \mathbf{J} &= -\mathbf{J_v}^{-1}\mathbf{J_u} \\ &= -\begin{bmatrix} 1/2l_1 & 0 \\ 0 & 1/2l_2 \end{bmatrix} \begin{bmatrix} -2(x - x_A) & -2(y - y_A) \\ -2(x - x_B) & -2(y - y_B) \end{bmatrix} \\ &= \begin{bmatrix} (x - x_A)/l_1 & (y - y_A)/l_1 \\ (x - x_B)/l_2 & (y - y_B)/l_2 \end{bmatrix} \end{aligned} \tag{D.6}$$

The inverse of $\mathbf{J}$ is

$$\mathbf{J}^{-1} = \frac{1}{\det(\mathbf{J})} \begin{bmatrix} (y - y_B)/l_2 & -(y - y_A)/l_1 \\ -(x - x_B)/l_2 & (x - x_A)/l_1 \end{bmatrix} \tag{D.7}$$

where the determinant of $\mathbf{J}$ is

$$\det(\mathbf{J}) = \frac{(x-x_A)(y-y_B)-(x-x_B)(y-y_A)}{l_1 l_2} \tag{D.8}$$

The condition number κ of the Jacobian is given by $\kappa = \|\mathbf{J}\|\|\mathbf{J}^{-1}\|$ (equation (4.19)), where the norm of $\mathbf{J}$ is defined as $\|\mathbf{J}\| = [\mathrm{tr}(\mathbf{J}\mathbf{W}\mathbf{J}^{\mathsf{T}})]^{\frac{1}{2}}$ (equation (4.20)). In order to find the norm of $\mathbf{J}$, we first need

$$\begin{aligned}\mathbf{J}\mathbf{W}\mathbf{J}^{\mathsf{T}} &= \begin{bmatrix} \frac{x-x_A}{l_1} & \frac{y-y_A}{l_1} \\ \frac{x-x_B}{l_2} & \frac{y-y_B}{l_2} \end{bmatrix} \begin{bmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{bmatrix} \begin{bmatrix} \frac{x-x_A}{l_1} & \frac{x-x_B}{l_2} \\ \frac{y-y_A}{l_1} & \frac{y-y_B}{l_2} \end{bmatrix} \\ &= \begin{bmatrix} \frac{(x-x_A)^2+(y-y_A)^2}{2l_1^2} & \cdot \\ \cdot & \frac{(x-x_B)^2+(y-y_B)^2}{2l_2^2} \end{bmatrix}\end{aligned} \tag{D.9}$$

Thus the norm is given by

$$\begin{aligned}\|\mathbf{J}\| &= [\mathrm{tr}(\mathbf{J}\mathbf{W}\mathbf{J}^{\mathsf{T}})]^{\frac{1}{2}} \\ &= \left[\frac{(x-x_A)^2+(y-y_A)^2}{2l_1^2} + \frac{(x-x_B)^2+(y-y_B)^2}{2l_2^2}\right]^{\frac{1}{2}} \\ &= \left[\frac{l_1^2}{2l_1^2} + \frac{l_2^2}{2l_2^2}\right]^{\frac{1}{2}} \\ &= 1\end{aligned} \tag{D.10}$$

In a similar way it can be shown that

$$\mathbf{J}^{-1}\mathbf{W}\mathbf{J}^{-1\mathsf{T}} = \frac{1}{2(\det(\mathbf{J}))^2} \begin{bmatrix} \frac{y-y_B}{l_2} + \frac{y-y_A}{l_1} & \cdot \\ \cdot & \frac{x-x_B}{l_2} + \frac{x-x_A}{l_1} \end{bmatrix} \tag{D.11}$$

where $\det(\mathbf{J})$ is given by (D.8). The norm of $\mathbf{J}^{-1}$ is thus

$$\begin{aligned}\|\mathbf{J}^{-1}\| &= [\mathrm{tr}(\mathbf{J}^{-1}\mathbf{W}\mathbf{J}^{-1\mathsf{T}})]^{\frac{1}{2}} \\ &= \frac{1}{\det(\mathbf{J})}\end{aligned} \tag{D.12}$$

Substituting (D.10) and (D.12) into (4.19), the condition number κ is given by

$$\begin{aligned}\kappa &= \|\mathbf{J}\|\|\mathbf{J}^{-1}\| \\ &= \frac{1}{\det(\mathbf{J})}\end{aligned} \tag{D.13}$$

The *inverse* condition number is thus

$$\begin{aligned} \kappa^{-1} &= \det(\mathbf{J}) \\ &= \frac{(x-x_A)(y-y_B)-(x-x_B)(y-y_A)}{l_1 l_2} \end{aligned} \tag{D.14}$$

Assuming that $y_B = y_A$, (D.14) can be simplified to

$$\kappa^{-1} = \frac{(y-y_A)(x_B-x_A)}{l_1 l_2} \tag{D.15}$$

The necessary condition for a maxima or minima is $\nabla\kappa^{-1} = 0$. The partial derivative of κ^{-1} with respect to x is

$$\begin{aligned} \frac{\partial\kappa^{-1}}{\partial x} &= -\frac{(y-y_A)(x_B-x_A)}{l_1^3 l_2^3}\left[(x-x_A)l_2^2+(x-x_B)l_1^2\right] \\ &= -\frac{(x_B-x_A)}{l_1^3 l_2^3}(y-y_A)(2x-x_A-x_B) \\ &\quad \times[(x-x_A)(x-x_B)+(y-y_A)^2] \end{aligned} \tag{D.16}$$

Assuming $(x_B - x_A) > 0$ for all that follows, there are three cases for which $\frac{\partial\kappa^{-1}}{\partial x} = 0$.

Case X1:

$$\begin{aligned} y - y_A &= 0 \\ \therefore y &= y_A \end{aligned} \tag{D.17}$$

Case X2:

$$\begin{aligned} 2x - x_A - x_B &= 0 \\ \therefore x &= \tfrac{x_A+x_B}{2} \end{aligned} \tag{D.18}$$

Case X3:

$$\begin{aligned} (x-x_A)(x-x_B)+(y-y_A)^2 &= 0 \\ \therefore \left[x-\left(\tfrac{x_A+x_B}{2}\right)\right]^2+(y-y_A)^2 &= \left(\tfrac{x_A-x_B}{2}\right)^2 \end{aligned} \tag{D.19}$$

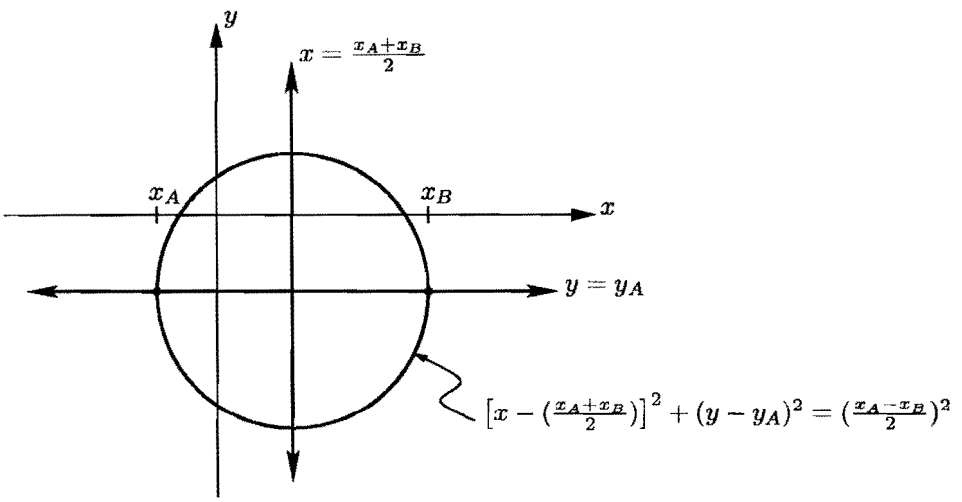

Figure D.1: Solutions to $\frac{\partial \kappa^{-1}}{\partial x} = 0$

These three cases are depicted graphically in Figure D.1.

The partial derivative of κ^{-1} with respect to y is

$$\begin{aligned}\frac{\partial \kappa^{-1}}{\partial y} &= -\frac{(x_B - x_A)}{l_1^3 l_2^3}\left[l_1^2 l_2^2 - (y-y_A)^2 l_2^2 - (y-y_A)^2 l_1^2\right] \\ &= -\frac{(x_B - x_A)}{l_1^3 l_2^3}[(x-x_A)(x-x_B) + (y-y_A)^2] \\ &\quad \times[(x-x_A)(x-x_B) - (y-y_A)^2]\end{aligned} \tag{D.20}$$

There are two cases for which $\frac{\partial \kappa^{-1}}{\partial y} = 0$.

Case Y1:

$$\begin{aligned}&(x-x_A)(x-x_B) + (y-y_A)^2 = 0 \\ \therefore &\left[x - \left(\tfrac{x_A+x_B}{2}\right)\right]^2 + (y-y_A)^2 = \left(\tfrac{x_A-x_B}{2}\right)^2\end{aligned} \tag{D.21}$$

Case Y2:

$$\begin{aligned}&(x-x_A)(x-x_B) - (y-y_A)^2 = 0 \\ &\therefore y = \pm\sqrt{(x-x_A)(x-x_B)} + y_A\end{aligned} \tag{D.22}$$

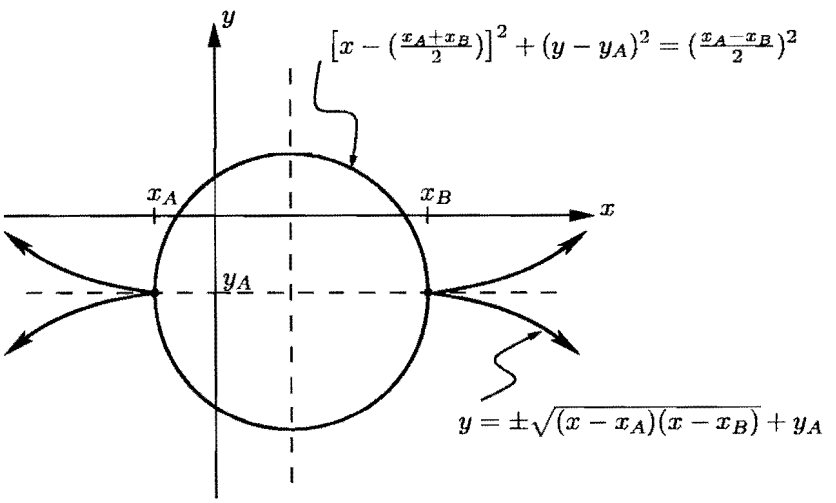

Figure D.2: Solutions to $\frac{\partial \kappa^{-1}}{\partial y}=0$

These two cases are depicted in Figure D.2. Note that for case Y2, real values of y from (D.22) only occur when $x \leq x_A$ or $x \geq x_B$. Comparing the above results it is evident that both necessary conditions are only simultaneously satisfied by the set of points U corresponding to (D.19) and (D.21) from cases X3 and Y1:

$$\mathbf{U}=\left\{\mathbf{u}\in\Re^2:\left[x-\left(\frac{x_A+x_B}{2}\right)\right]^2+(y-y_A)^2=\left(\frac{x_A-x_B}{2}\right)^2\right\} \quad \text{(D.23)}$$

which is the set of points on a circle of radius $(x_A-x_B)/2$ centered at $((x_A+x_B)/2, y_A)$.

Transform equation (D.15) to polar coordinates (r,θ) by substituting $y=y_A+r\sin\theta$ and $x=(x_A+x_B)/2+r\cos\theta$:

$$\kappa^{-1}=\frac{(x_B-x_A)r\sin\theta}{\left[\left(\frac{x_B-x_A}{2}+r\cos\theta\right)^2+r^2\sin^2\theta\right]^{\frac{1}{2}}\left[\left(-\frac{x_B-x_A}{2}+r\cos\theta\right)^2+r^2\sin^2\theta\right]^{\frac{1}{2}}} \quad \text{(D.24)}$$

Squaring both sides of equation (D.24) and simplifying yields

$$(\kappa^{-1})^2=\frac{4r^2\left(\frac{x_B-x_A}{2}\right)^2-4r^2\left(\frac{x_B-x_A}{2}\right)^2\cos^2\theta}{\left[\left(\frac{x_B-x_A}{2}\right)^2+r^2\right]^2-4\left(\frac{x_B-x_A}{2}\right)^2r^2\cos^2\theta} \quad \text{(D.25)}$$

It is evident that when $r = (x_B - x_A)/2$, which corresponds to the set of points $\mathbf{U}$, that $(\kappa^{-1})^2 = 1$. It follows that $\kappa^{-1} = \pm 1$. However, $\kappa^{-1} \in [0, 1]^\mathsf{T}$, thus $\kappa^{-1} = 1$ and the set $\mathbf{U}$ corresponds to a "maximum ridge".

Assume that $y > y_A$ and that there exists a point $\mathbf{u}'$ *inside* the prescribed workspace W_p such that $\kappa^{-1}(\mathbf{u}')$ is a minima. It follows that $\nabla\kappa^{-1}(\mathbf{u}') = \mathbf{0}$. However it has been shown that the only solution to $\nabla\kappa^{-1}(\mathbf{u}') = \mathbf{0}$ is $\mathbf{U}$, and this set of points corresponds to a maxima. It thus follows that the minimum value of κ^{-1} must lie on the boundary ∂W_p of the workspace W_p.

Appendix E

Minimum norm solution of a set of equations

THEOREM. *On removal of inequality constraints, equation (6.17) is equivalent to (6.14)*

PROOF. In optimization terms, the minimum norm solution of (6.11) may be stated as

$$\min_{\mathbf{t}} \frac{1}{2}\mathbf{t}^{\mathsf{T}}\mathbf{t} \quad \text{such that } \mathbf{S}\mathbf{t} + \mathbf{f}^{P} = \mathbf{0} \tag{E.1}$$

where $\mathbf{S}$ is an $m \times n$ matrix with $m < n$. This optimization problem is equivalent to (6.17) with inequality constraints removed.

The Lagrangian $\mathcal{L}$ of this optimization problem is

$$\mathcal{L}(\mathbf{t}, \boldsymbol{\lambda}) = \frac{1}{2}\mathbf{t}^{\mathsf{T}}\mathbf{t} + \boldsymbol{\lambda}^{\mathsf{T}}(\mathbf{S}\mathbf{t} + \mathbf{f}^{P}) \tag{E.2}$$

where $\boldsymbol{\lambda} \in \Re^{m}$ denotes the vector of Lagrange multipliers.

The necessary conditions for a minimum are:

$$\nabla_{t}\mathcal{L} = \mathbf{0} \tag{E.3}$$

Applying these to (E.2) yields

$$\mathbf{t} + (\boldsymbol{\lambda}^{\mathsf{T}}\mathbf{S})^{\mathsf{T}} = \mathbf{0} \tag{E.4}$$

$$\therefore \quad \mathbf{t} + \mathbf{S}^{\mathsf{T}}\boldsymbol{\lambda} = \mathbf{0} \tag{E.5}$$

$$\therefore \quad \mathbf{S}\mathbf{t} + \mathbf{S}\mathbf{S}^{\mathsf{T}}\boldsymbol{\lambda} = \mathbf{0} \tag{E.6}$$

$$\therefore \quad \boldsymbol{\lambda} = -(\mathbf{S}\mathbf{S}^{\mathsf{T}})^{-1}\mathbf{S}\mathbf{t} \tag{E.7}$$

By substituting (E.7) into (E.5) it follows that

$$\mathbf{t} - \mathbf{S}^{\mathsf{T}}(\mathbf{S}\mathbf{S}^{\mathsf{T}})^{-1}\underbrace{\mathbf{S}\mathbf{t}}_{-\mathbf{f}^{P}} = \mathbf{0} \tag{E.8}$$

$$\therefore \quad \mathbf{t} = -\mathbf{S}^{\mathsf{T}}(\mathbf{S}\mathbf{S}^{\mathsf{T}})^{-1}\mathbf{f}^{P} \tag{E.9}$$

which is the minimum norm solution given in (6.14).