Chapter 1

Introduction

The design of robotic systems involves the development of mathematical models for real-time analysis. The robot under consideration, called Ursula, was designed by the Robotics and Mechanisms Group, Department of Mechanical Engineering, Virginia Tech, in collaboration with B&W Nuclear Technologies for the purpose of weld inspection in a nuclear reactor vessel which is filled with water. This design aims at reducing radiation exposure hazards, maintenance and transportation costs.

Ursula is a manipulator with six degrees of freedom, attached to a mobile base. Ursula was designed with the capability of traversing all along the inner circumference of the reactor vessel. Suction cups hold its base to the vessel wall during the weld inspection process. Ursula being smaller compared to the side of the reactor vessel, it repositions itself and thus continues scanning new weldments. Ursula is a mobile robot and hence it necessitates the use of a control system which would continuously track its *pose*, that is, its position and orientation. The pose of the Ursula needs to be determined in terms of the reactor vessel coordinate system which is considered in the present setup as the global coordinate system or the global positioning system in which all the objects are referred. The sketch of Ursula performing weldment inspections in the nuclear reactor vessel and the reactor vessel coordinate system are shown in Figure 1.1.

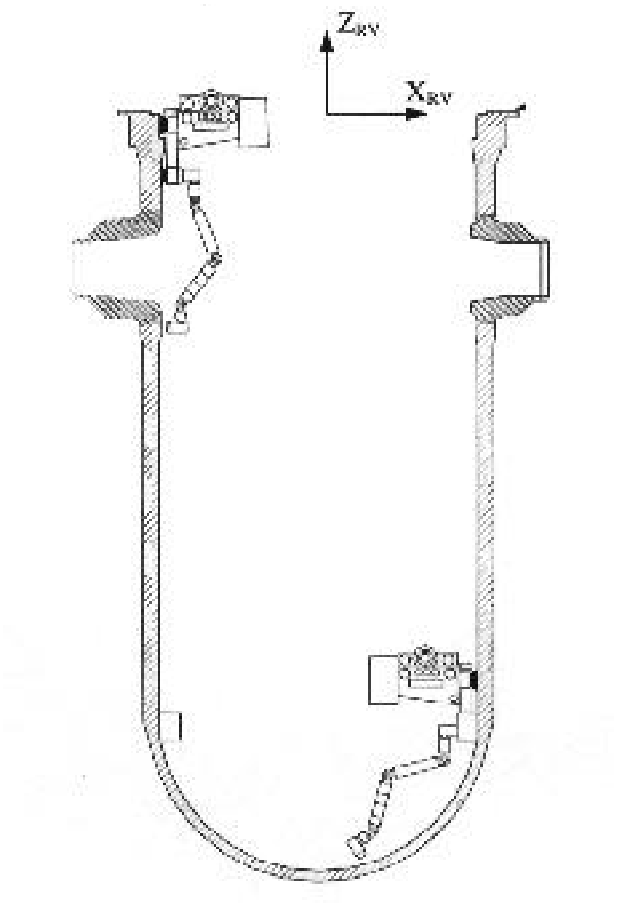


Figure 1.1: Two Ursula Robots in the Reactor Vessel

The Robotics and Mechanisms Group has developed a unique hardware setup to solve the problem of finding the pose of Ursula. In this setup, a laser gun, having two degrees of freedom is mounted at the center of a beam which runs across the flange of the reactor vessel. A photosensor target is also placed at the base of Ursula. The setup also includes four other photosensor targets that are placed at known locations in the reactor vessel for the purpose of laser gun calibration. The laser gun calibration and Ursula's global pose calculation procedures are discussed below.

The laser gun mounted on two revolute joints that provide two rotational degrees of freedom along mutually perpendicular axes, i.e., the pan axis and the tilt axis. With these two rotary movements, it is possible to aim the laser gun at any point in the reactor vessel. The laser gun calibration procedure involves finding the laser gun's global pose with the help of the four targets placed at known locations in the reactor vessel. Accordingly, the laser gun is directed to find and shoot the targets, one at a time. Each time, the joint angles are read from the rotation encoders located on the revolute joints. The global pose of the laser gun then needs to be determined by relating the global pose of the targets and the rotation encoder readings. Once the laser gun's global pose is determined, the laser gun is then directed to shoot the target located at the base of Ursula. The global pose of Ursula then needs to be determined by relating the radius of the reactor vessel, the depth gauge reading and the rotation encoder readings obtained from the laser gun's revolute joints and Ursula's base. Thus, the location of Ursula can be found with the hardware setup described above.

The pose of Ursula is sensitive to the kinematic errors in the setup. These errors are errors in the object geometries and hence are also known as geometric errors. The global pose of Ursula needs to be found while taking these inaccuracies into account.

The present analytical procedures to find the pose of Ursula do not involve suitable optimization algorithms that result in a reliable or robust convergence performance. Moreover, modeling and solving the problem by taking advantage of its special structure would result in a better and quicker resolution of this problem. The present research intends to model the problem effectively and to investigate various deflected gradient based search algorithms for finding the pose of Ursula.

The hardware setup and the pose determination problem of Ursula have been presented above. Chapter 2 now gives a review of the available literature on pose determination problems along with applicable optimization procedures. The modeling aspects of the problem to represent the setup of the system as described above are discussed in Chapter 3. The concept of frames and transformations needed in the formulation of the problem is explained in this context. The techniques for solving the problems modeled in Chapter 3 are discussed in Chapter 4. Modifications made to various existing nonlinear programming procedures are explained in this context. A description of the test problems used to validate the model and the prescribed solution procedures is also presented. Some implementation guidelines and results are discussed in Chapter 5. Chapter 6 summarizes the present research efforts and suggests ideas for further research in this field. Finally, Chapter 7 lists the references of the literature reviewed.

Chapter 2

Literature Review

This chapter aims at providing a brief overview of pose determination and robotic control and calibration procedures. It also discusses the application of nonlinear least squares and other optimization problems in the area of robotics and controls. This chapter focuses on the present research efforts and the existing literature.

2.1 Design and Development of Ursula

Robotic design has been widely researched and suitable design methodologies have been developed for many specific environments. Reinholtz, Shooter, Fallon and Glass (1994) discuss the development efforts of an underwater robotic system which includes the conceptual design, operation plan, hardware setup and the global positioning of Ursula. An acoustics based pose determination system for Ursula was discussed by Tidwell et al.(1993). This system was later replaced by the existing laser based positioning system. Voruganti (1995), in his dissertation, presents the development and application of a robot system characterization methodology which improves the performance of Ursula. This methodology involves modeling, identification, analysis and minimization of errors in Ursula's geometric parameters. The principles of robotics, the concepts of frames and transformations and the link representations have been described in Craig (1984).

2.2 Optimization Techniques in Robotic Control and Calibration

Optimization problems arise in numerous real-life applications, many of them related to control. Optimization and simulation techniques have been extensively applied to problems in robotics in the areas of inverse kinematics and calibration. Zhuang, Wang and Roth (1994) discuss the application of the simulated annealing approach to obtain optimal measurements for robot calibration. This includes the measurement of position and orientation of the robot end-effector world coordinates (chosen reference system). Their approach results in a sufficiently accurate robotic calibration, but suffers from the disadvantage that it is computationally expensive. Han, Snyder and Bilbro (1990) suggest a tree annealing algorithm, an extension to the simulated annealing algorithm, for determining the position and orientation of analytic surfaces. Their methodology performs well even in the presence of considerable amounts of noise in the range images. Stone and Sanderson (1988) utilize Monte-Carlo simulation techniques for the generation of positioning and orienting errors. Zak et al. (1991) employ a weighted least squares approach to robot kinematic parameter estimation. They later (1993) developed a simulation technique for the improvement of robotic calibration. Law and Kelton (1991), in their text, describe an algorithm for the generation of random numbers that are characterized by good uniformity and reproducibility. Optimization techniques have also been applied to robotic design problems, such as inverse kinematics and collision detection [Gallerini and Sciomachen, (1993)].

2.3 Nonlinear Least Squares Problems

Nonlinear least squares (NLS) problems, a class of nonlinear optimization problems bearing a special structure, have evoked a great deal of research interest, resulting in an extensive discussion in the literature. These problems find applications in contexts ranging from curve fitting to neural networks. Relevant discussion on such problems and solution methodologies can be found in the book by Dennis and Schnabel (1983). Gorinevsky (1994) proposed an algorithm for on-line parametric nonlinear least squares optimization. Nobiki et al. (1993) use the least squares minimization technique in an estimation method for an accurate multi-viewpoint stereo measurement.

2.4 Nonlinear Optimization Techniques

Nonlinear programming techniques include various derivative-based and derivative-free techniques that are capable of performing an iterative search in the multidimensional space [Bazaraa, Sherali, Shetty (1993), Luenberger (1984), Nash and Sofer (1996)]. These techniques are applicable for unconstrained optimization problems. Constrained optimization problems can be solved using appropriate barrier functions used in tandem with the objective function. Modifications of multidimensional search techniques to accommodate simple constraints have also been considered (see Bazaraa et al., 1993).

The optimization procedures mentioned earlier, guarantee stationary point convergence under mild assumptions. Some of these procedures converge in a finite number of steps equal to the number of variables, if the objective function is quadratic, and yield superlinear convergence rates for nonquadratic functions under certain assumptions. Pendyala (1994), in his masters thesis, has investigated several nonlinear programming techniques to solve a design problem in the field of wireless communication. The response surface methodology technique, which is a simulation optimization technique, has also been used.

The texts by Bazaraa et al.(1993), Luenberger (1984), Jacoby et al. (1972) and Nash and Sofer (1996) provide a good discussion of various iterative nonlinear optimization techniques.

This research effort is expected to contribute toward developing effective solution methodologies in the area of robotic design and control that require the optimization of highly nonlinear and complex functions whose behavior is largely unknown.

Chapter 3

Modeling the Problem

In this chapter, we develop a mathematical model for determining Ursula's pose. An introduction to the concepts of reference frames and transformations is also presented in this chapter.

3.1 Reference Frames and Transformations

The relative position and orientation of the associated links in the laser gun mechanism can be described by attaching reference frames to them. A frame is a set of four vectors that describe its pose, i.e., position and orientation with respect to the parent frame [Craig, J.J (1984)]. A frame B, represented by [B], could be described relative to another frame [A] by,

$$\{ {}^{A}R_{B}, {}^{A}P_{Borg} \} \tag{3.1}$$

where, $\{{}^{A}R_{B}\}$ describes the orientation of frame B relative to Frame A and $\{ {}^{A}P_{Borg}\}$ describes the position of frame B's origin relative to that of frame A. Moreover, a frame B defined relative to frame A can be equivalently defined with respect to frame C using the transformation

$$\{{}^{C}R_{B}\} = [{}^{C}T_{A}][{}^{A}R_{B}] \qquad (3.2)$$

where $[{}^{C}T_{A}]$, is a homogenous transform of frame A about frame C, and can be constructed as follows. If α_{CA}, β_{CA} and γ_{CA} are the yaw, pitch and roll angles, i.e., the angles of rotation made by frame A relative to frame C about the Z, Y and X axes respectively, and x_{Aorg}, y_{Aorg} and z_{Aorg} represent the location of frame A's origin with respect to frame C, the transform $[{}^{C}T_{A}]$ is given by

$$\begin{bmatrix} \cos a_{CA}\cos b_{CA} & \cos a_{CA}\sin b_{CA}\sin g_{CA} - \sin a_{CA}\cos g_{CA} & \cos a_{CA}\sin b_{CA}\cos g_{CA} + \sin a_{CA}\sin g_{CA} & x_{Aorg} \\ \sin a_{CA}\cos b_{CA} & \sin a_{CA}\sin b_{CA}\sin g_{CA} + \cos a_{CA}\cos g_{CA} & \sin a_{CA}\sin b_{CA}\cos g_{CA} - \cos a_{CA}\sin g_{CA} & y_{Aorg} \\ -\sin b_{CA} & \cos b_{CA}\sin g_{CA} & \cos b_{CA}\cos g_{CA} & z_{Aorg} \\ 0 & 0 & 0 & 1 \end{bmatrix}. \qquad (3.3)$$

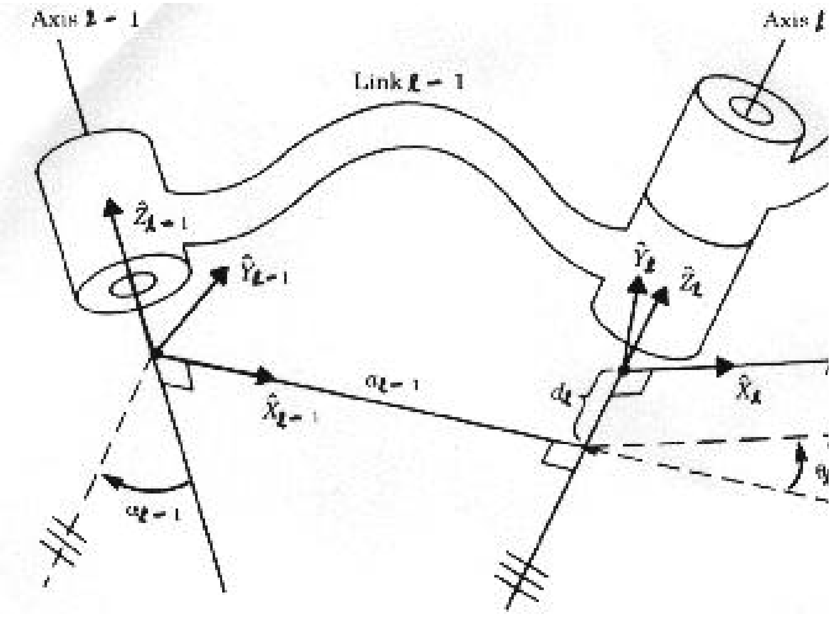


Figure 3.1: Representation of the Link Parameters (Denavit-Hartenberg Notation)

Using the Denavit-Hartenberg notation [Craig, J.J. (1984)], the frame attached to link l-1 shown in Figure 3.1 can be represented as follows. Using the link parameters namely, the link offset $d_{l\text{-}1}$, the link length $a_{l\text{-}1}$, and the relative link parameters such as the link twist a_l, and the joint angle y_l, the homogenous transform [${}^{A}T_{B}$] can be written as

$$\begin{bmatrix} \cos\psi_l & \sin\psi_l & 0 & a_{l-1} \\ \sin\psi_l\cos\alpha_l & \cos\psi_l\cos\alpha_{il} & -\sin\alpha_l & -d_{l-1}\sin\alpha_l \\ \sin\psi_l\sin\alpha_l & \cos\psi_l\sin\alpha_l & \cos\alpha_l & d_{il-1}\cos\alpha_l \\ 0 & 0 & 0 & 1 \end{bmatrix}. \qquad (3.4)$$

Frames attached to the laser gun mechanism

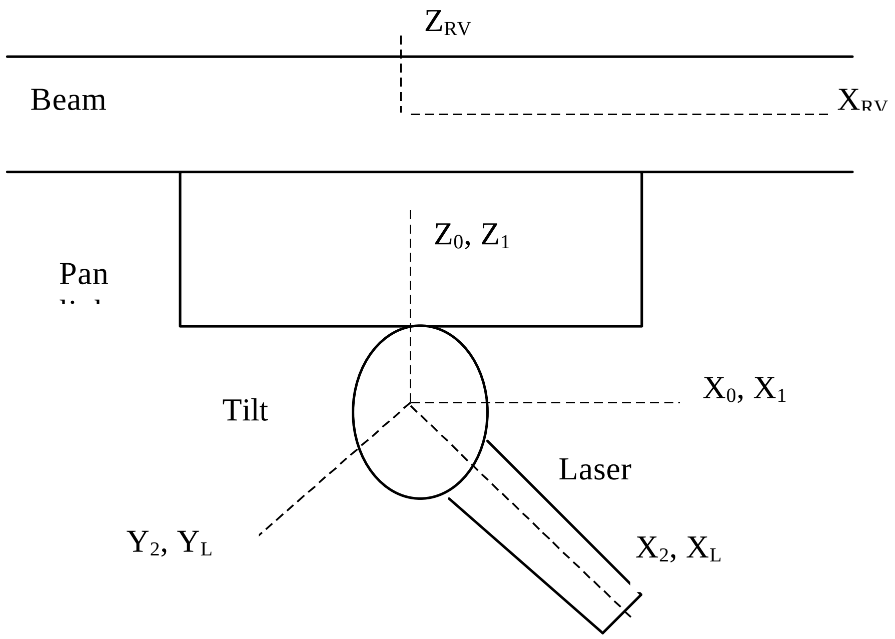


Figure 3.2: Laser Gun Mechanism’s Coordinate System

A sketch of the laser gun mechanism is shown in Figure 3.2. A frame ‘RV’ is attached to the reactor vessel whose coordinate system given by $\{X_{RV}, Y_{RV}, Z_{RV}\}$, represents the global positioning system. The pose of Ursula has to be determined relative to this system. Considering the laser gun mechanism as a unit, a frame ‘0’ is attached to it. It can be noted that the assignment of frames is a matter of convenience. As discussed earlier, the laser gun mechanism is comprised of a laser gun and a two of revolute joints. These can be considered as individual links. Hence, a frame ‘1’ is attached to the pan axis, a frame ‘2’ is attached to the tilt axis, and finally, a frame ‘L’ is attached to the laser gun.

3.2 Mathematical Modeling of the Problem

When the laser gun shoots a photosensor target, this target would be referred to in the frame of the laser gun, which is frame ‘L’. The position of the target in that frame, ${}^{L}P$, would be given by $\{x_L, 0, 0\}$, as it would be along the X axis (axis of the barrel) of the frame ‘L’. This position can be expressed relative to the reactor vessel frame ‘RV’ through a series of transformations. That is to say, the position of the target in the reactor vessel coordinate system, ${}^{RV}P$, would be given by

$$ {}^{RV}P = [{}^{RV}T_{0}]\,[{}^{0}T_{1}][{}^{1}T_{2}][{}^{2}T_{L}][{}^{L}P]. \qquad (3.5) $$

The vector ${}^{RV}P$ and the transformation matrices $[{}^{RV}T_{0}]$, $[{}^{0}T_{1}]$, $[{}^{1}T_{2}]$ and $[{}^{2}T_{L}]$ can be constructed in the following manner.

If the target is located in the reactor vessel at {x_T, y_T, z_T}, then the vector ${}^{RV}P$ is given by

$$ {}^{RV}P = \begin{bmatrix} x_T \\ y_T \\ z_T \\ 1 \end{bmatrix}. \tag{3.6} $$

If {x, y, z, α, β, γ}are the pose parameters of the laser gun mechanism relative to the frame RV, then the transformation matrix [${}^{RV}T_0$] is given by

$$ \begin{bmatrix} \cos\alpha\cos\beta & \cos\alpha\sin\beta\sin\gamma - \sin\alpha\cos\gamma & \cos\alpha\sin\beta\cos\gamma + \sin\alpha\sin\gamma & x \\ \sin\alpha\cos\beta & \sin\alpha\sin\beta\sin\gamma + \cos\alpha\cos\gamma & \sin\alpha\sin\beta\cos\gamma - \cos\alpha\sin\gamma & y \\ -\sin\beta & \cos\beta\sin\gamma & \cos\beta\cos\gamma & z \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{3.7} $$

If θ_I and θ are the respective initial and final pan joint angles read from the rotation encoder on the pan joint, a_0 is the pan link length, d_1 is the pan link offset, and α_0 is the pan link twist angle, using the Denavit-Hartenberg notation [Craig, J.J., (1984)], the transformation matrix [${}^0T_1$] is given by

$$ \begin{bmatrix} \cos\theta_I\cos\theta & \sin\theta_I\cos\theta + \cos\theta_I\sin\theta & 0 & a_0 \\ (\sin\theta_I\cos\theta + \cos\theta_I\sin\theta)\cos\alpha_0 & (\cos\theta_I\cos\theta - \sin\theta_I\sin\theta)\cos\alpha_0 & -\sin\alpha_0 & -d_1\sin\alpha_0 \\ (\sin{}_I\cos\theta + \cos{}_I\sin\theta)\sin\alpha_0 & (\cos\theta_I\cos\theta - \sin\theta_I\sin\theta)\sin\alpha_0 & \cos\alpha_0 & d_1\cos\alpha_0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{3.8} $$

Similarly, for the tilt link, with link parameters given by ϕ_I and ϕ (read from the rotation encoder on the tilt joint), a_1, d_2 and α_1, the transformation matrix $[^1 T_2]$ using the Denavit-Hartenberg notation, is given by

$$\begin{bmatrix} \cos\phi_I \cos\phi & \sin\phi_I \cos\phi + \cos\phi_I \sin\phi & 0 & a_1 \\ (\sin\phi_I \cos\phi + \cos\phi_I \sin\phi)\cos\alpha_1 & (\cos\phi_I \cos\phi - \sin\phi_I \sin\phi)\cos\alpha_1 & -\sin\alpha_1 & -d_2 \sin\alpha_1 \\ (\sin\phi_I \cos\phi + \cos\phi_I \sin\phi)\sin\alpha_1 & (\cos\phi_I \cos\phi - \sin\phi_I \sin\phi)\sin\alpha_1 & \cos\alpha_1 & d_2 \cos\alpha_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

(3.9)

If $\{x_L, y_L, z_L, \alpha_L, \beta_L, \gamma_L\}$ are the pose parameters of the laser gun with respect to the laser gun mechanism (as a complete unit), then the transformation matrix $[^2 T_L]$ is given by

$$\begin{bmatrix} \cos\alpha_L \cos\beta_L & \cos\alpha_L \sin\beta_L \sin\gamma_L - \sin\alpha_L \cos\gamma_L & \cos\alpha_L \sin\beta_L \cos\gamma_L + \sin\alpha_L \sin\gamma_L & x_L \\ \sin\alpha_L \cos\beta_L & \sin\alpha_L \sin\beta_L \sin\gamma_L + \cos\alpha_L \cos\gamma_L & \sin\alpha_L \sin\beta_L \cos\gamma_L - \cos\alpha_L \sin\gamma_L & y_L \\ -\sin\beta_L & \cos\beta_L \sin\gamma_L & \cos\beta_L \cos\gamma_L & z_L \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

(3.10)

The position of the target in the laser gun frame ‘L’, as mentioned earlier, is given by

$$\begin{bmatrix} p \\ 0 \\ 0 \\ 1 \end{bmatrix}. \qquad (3.11)$$

The parameters that vary with the position of the target include the pan joint angle, the tilt joint angle, and the distance of the target from the tip of the laser gun along the X axis, which is also the laser length and is given by x_L. The pan and tilt joint angles can be read from the rotation encoders located on the respective joints, but the laser length cannot be measured. Therefore, the laser length x_L remains as an unknown parameter along with the laser gun's pose parameters. The nominal values of the other pan and tilt link parameters are known. Therefore, the transforms given by Equations (3.8) and (3.9) can be evaluated. Substituting these in Equation (3.5), three equations relating the target's position, the pose parameters of the laser gun mechanism and the laser length x_L, can be obtained. To solve for the six pose parameters of the laser gun mechanism would therefore require three targets. Due to the presence of kinematic errors in the setup, a fourth target is used to determine the values of the pose parameters more accurately.

For each target *i*, the product of the transforms $[{}^0 T_1]$, $[{}^1 T_2]$, $[{}^2 T_L]$ and $[{}^L P]$ may be written as

$$\begin{bmatrix} g_{i1}p_i + h_{i1} \\ g_{i2}p_i + h_{i2} \\ g_{i3}p_i + h_{i3} \\ 1 \end{bmatrix} \forall i = 1,2,3,4 \qquad (3.12)$$

where p_i is the laser length for target i=1,2,3,4, and where the values of g_{ij} and h_{ij} ($\forall$i=1,2,3,4 and $\forall$j = 1,2,3) are given by appropriate functions of the parameters θ_i, θ, α_0, a_0, d_1, ϕ_i, ϕ, α_1, a_1, d_2, x_L, y_L, z_L, α_L, β_L and γ_L as dictated by Equations (3.8), (3.9) and (3.10). The nominal values and error ranges of these parameters are known. Thus, the values of g_{ij} and h_{ij} are essentially known.

The transform $[{}^{RV}T_0]$ may be partitioned as

$$\left[\begin{array}{ccc:c} & & & x \\ & A_\Pi & & y \\ & & & z \\ \hdashline 0 & 0 & 0 & 1 \end{array}\right]. \qquad (3.13)$$

The vector on the left-hand side of (3.5) is the location of the target *i* in the reactor vessel coordinate system, denoted by ${}^{RV}P_i$, and given by Equation (3.6) subscripted by *i*. Therefore, the equation system (3.5) may be written as,

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} + \left[A_\Pi\right] \begin{bmatrix} p_i g_{i1} \\ p_i g_{i2} \\ p_i g_{i3} \end{bmatrix} + \left[A_\Pi\right] \begin{bmatrix} h_{i1} \\ h_{i2} \\ h_{i3} \end{bmatrix} = \begin{bmatrix} x_{Ti} \\ y_{Ti} \\ z_{Ti} \end{bmatrix} \forall i = 1,2,3,4\,. \qquad (3.14)$$

This equation can be further simplified to

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} x_{i1}p_i \\ x_{i2}p_i \\ x_{i3}p_i \end{bmatrix} = \begin{bmatrix} z_{i1} \\ z_{i2} \\ z_{i3} \end{bmatrix} \forall i = 1,2,3,4 \,, \tag{3.15}$$

where the value of ξ_{ij} is given by a function of α, β, γ and g_{ij} (for $\forall i$=1,2,3,4 and j=1,2,3). The value of ζ_{ij} is given by a function of α, β, γ, h_{ij}, x_{Ti}, y_{Ti} and z_{Ti} (for $\forall i$=1,2,3,4 and j=1,2,3). As α, β and γ are unknown parameters, the values of ξ_{ij} and ζ_{ij} are unknown.

Thus, for four targets, i.e. i = 1,2,3 and 4, twelve equations can be obtained by equating the row elements of (3.15). Eliminating x, y and z from these equations for i =1, for example, nine equations involving the parameters α, β, γ and p_i i=1,2,3,4, can be obtained. These are given as follows.

$$\begin{bmatrix} x_{11}p_1 \\ x_{12}p_1 \\ x_{13}p_1 \end{bmatrix} - \begin{bmatrix} x_{i1}p_i \\ x_{i2}p_i \\ x_{i3}p_i \end{bmatrix} = \begin{bmatrix} z_{11} - z_{i1} \\ z_{12} - z_{i2} \\ z_{13} - z_{i3} \end{bmatrix} \forall i = 2,3,4 \,. \tag{3.16}$$

Using the least squares technique, an objective function may be formulated as,

$$Minimize \sum_{i=2}^{4} \sum_{j=1}^{3} \left(\left(x_{1j}p_1 - x_{ij}p_i \right) - \left(z_{1j} - z_{ij} \right) \right)^2 \,. \tag{3.17}$$

It is assumed in this context that the location of all the targets is known with equal confidence. Else, a weighted least squares objective function may be used. Moreover, the intent of this objective function is to try and find a laser gun pose which satisfies all the kinematic equations while minimizing the sum of squares of the errors.

Model for the laser gun pose determination problem

The laser gun mechanism's calibration problem can therefore be mathematically modeled as follows.

$$\begin{aligned} &\textit{Minimize} \quad f_1\,(\alpha, \beta, \gamma, p_1, p_2, p_3, p_4) \\ &\textit{subject to} \quad p_i \geq 0, \text{ for } i = 1,2,3,4 \end{aligned} \qquad (3.18)$$

where α = the yaw angle of rotation made by the laser gun relative to the reactor vessel

β = the pitch angle of rotation made by the laser gun relative to the reactor vessel

γ = the roll angle of rotation made by the laser gun relative to the reactor vessel

p_i = the unknown laser lengths when targets i = 1,2,3,4 are shot.

The function f_1 is represented by a least squares formulation given by (3.17), in which the above listed parameters are unknown. This objective function is highly nonlinear and its behavior is largely unknown. The solution procedures investigated to optimize such highly nonlinear functions are discussed in the next chapter. Optimizing the above function would hence result in the determination of the pose parameters of the laser gun mechanism. The values of x, y, z can then be found via the equations,

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} z_{11} \\ z_{12} \\ z_{13} \end{bmatrix} - \begin{bmatrix} x_{11}p_1 \\ x_{12}p_1 \\ x_{13}p_1 \end{bmatrix}. \tag{3.19}$$

Finding the pose of Ursula

Finding the pose of Ursula, using the above laser gun system, corresponds to the second phase of the problem solution. The location of Ursula may be conveniently expressed in the polar coordinate format, using the radius parameter *r*, the circumferential angle parameter j and the depth parameter *d*. In an attempt to simplify the problem and acquire the required information, a depth gauge that finds the depth *d* at which Ursula is located within the reactor vessel was physically installed. The rotation encoders present on Ursula give the orientation parameters. The reactor vessel has a circular cross section of radius *r*. Furthermore, Ursula is assumed to be in complete contact with the reactor vessel wall. To determine the pose of Ursula, therefore, the circumferential angle parameter j has to be determined. The following paragraph discusses a methodology to find this parameter.

Once the pose of the laser gun is determined, the photosensor target located on the tripod base of Ursula is aimed at and shot. Substituting for the pose of the laser gun, and the pan and tilt angles measured by the rotation encoders on the laser gun in the matrices given by (3.7), (3.8) and (3.9), respectively, and then substituting these matrices in (3.5), three simultaneous equations can be derived. It may be noted that there would be an additional unknown parameter p that represents the unknown laser length. Therefore, the two unknowns φ and p need to determined via these three equations. A suitable least squares objective function to accomplish this may be derived as follows. On substitution of the known laser gun pose and the rotation encoder readings in the matrices, Equation (3.5) may be reduced to

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} p \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} r\cos j \\ r\sin j \\ d \\ 1 \end{bmatrix}. \qquad (3.20)$$

The objective function can then be expressed as,

$$\textit{Minimize} \quad f_2\,(p,\ j), \qquad \textit{subject to} \qquad p^{3}\ 0 \qquad (3.21)$$

where, f_2 is given by $(a_{11}p + a_{14} - r\cos j)^2 + (a_{21}p + a_{24} - r\sin j)^2 + (a_{31}p + a_{34} - d)^2$.

An alternate approach

The problem of finding the location of Ursula can be reduced into a univariate problem. Reducing the number of variables in the problem would result in fewer iterations and function evaluations. The alternate equivalent formulation is developed below.

The function given by (3.21) can be projected into the j-space and solved as a univariate problem using line search techniques. This involves finding the optimal value of p for each value of j. Differentiating the function (3.21) partially with respect to p and setting the derivative equal to zero, a equation in terms of p is obtained. By the convexity of (3.21) in p, if the solution to this (linear) equation results in a nonnegative value of p, then that is the optimal value. Otherwise, the corresponding optimal value of p is zero. Thus, the objective function for this problem may be formulated as,

$$\text{Minimize} \qquad F_2\,(j) \qquad\qquad (3.22)$$

Note that $F_2\ (j\ )$ is evaluated via $f_2\ (p(j), j\ )$ where $p(j)$ is the optimum to (3.21) for a given j. The optimal p for (3.21), given by p*, is found according to the following, based on the optimum $j*$ determined for (3.22).

$$p^* = \begin{cases} \text{value of p that yeilds } \left[\dfrac{\P f_2\,(p,\ j\,*)}{\P p}\right] = 0, \text{ if this is nonnegative} \\ 0, \text{ otherwise.} \end{cases} \qquad (3.23)$$

3.3 Modeling for the Errors in the Setup

Kinematic errors exist in the laser gun mechanism’s hardware setup due to misalignments and machining inaccuracies. Therefore, the model has to suitably account for these errors. Based on the error sensitivity analysis made by Voruganti (1995), the parameters bearing error have been identified. The nominal values of these parameters are known along with the range of related errors. Errors in the setup can be simulated by randomly generating values within their ranges and adding them to the nominal values of the respective parameters. This procedure is discussed below.

Error modeling procedure

The procedure adopted for taking the errors in the setup into account for the laser gun calibration problem and for Ursula's pose determination problem includes the following steps.

1. Randomly generate the errors in the parameters α, β, γ, x, y, z, θ_I, α_0, d_1, ϕ_I, α_1, d_2, α_L, β_L, γ_L, x_L, y_L and z_L. Add these to their respective nominal values or theoretical values. The location of the targets is also a source of error. Therefore, generate the error in the parameters x_{Ti}, y_{Ti} and z_{Ti} (for i = 1,2,3,4) and add it to their nominal values.
2. Calculate theoretical values of θ_i and ϕ_i for each of the four targets (for i = 1,2,3,4) based on their location. These values correspond to the pan angle and tilt angle readings that would be read in the ideal case, i.e., when no errors are present in the parameters discussed in Step 1. The theoretical values of these parameters correspond to the starting solution of the problem discussed in Step 3.
3. Due to the errors present in the parameters discussed in Step 1, the pan and tilt angles required to shoot the targets are offset to a certain extent. Thus, to find the actual values of the pan and tilt angles, an initial optimization is required. Substituting the parameters

calculated in Step 1 in (3.7), (3.8), (3.9) and (3.10), an objective function similar to (3.17) may be obtained. The variables in this function are θ_i, ϕ_i and p_i (i=1,2,3,4). Minimizing this function, the actual values of θ_i and ϕ_i (i=1,2,3,4) may be obtained. Thus, these values of θ_i and ϕ_i (i=1,2,3,4) accommodate the kinematic errors in the hardware setup. The problem of determining the pose of the laser gun is similar to that discussed in this step. The objective function (3.17) is minimized to obtain the values of the pose parameters, instead of θ_i and ϕ_i (i=1,2,3,4). The procedure to accomplish this is discussed in the next step.

4. The pose parameters of the laser gun mechanism, given x, y, z, α, β and γ, are determined by solving the problem (3.18). The new values of θ_i and ϕ_i (i=1,2,3,4), calculated in Step 3 are used to compute the least squares objective function. The parameters which remain fixed at their nominal values while obtaining the solution to the problem are given by θ_I, α_0, d_1, ϕ_I, α_1, d_2, α_L, β_L, γ_L, x_L, y_L and z_L. Thus, the laser gun mechanism's pose is determined.
5. When the target on Ursula is shot, the rotation encoder angles θ and ϕ may be read. The depth at which Ursula is located in the vessel would be measured by a depth gauge. Add the randomly generated errors to these parameters. Note that Ursula moves along the circumference of the reactor vessel wall. The value of the parameter *r*, can be assumed to be given by the radius of the nuclear reactor vessel. Substitute these values while deriving Equation (3.20).

6. Substitute the calibrated laser gun's pose parameters, found in Step 4, along with the parameters given by θ_I, α_0, d_1, ϕ_I, α_1, d_2, α_L, β_L, γ_L, x_L, y_L and z_L. These are fixed at their nominal values as the error in the setup has already been accommodated in the calculation of the laser mechanism's pose parameters. Thus, Equation (3.20) may be derived.
7. The two remaining unknown parameters are the laser length p and the angle φ. Solve Problem (3.21) to obtain the optimal values of p and φ. Thus, the location of Ursula may be found.

In this chapter, the modeling aspects of the problem have been discussed. The solution methodologies applicable to execute the above procedure are presented in the next chapter.

Chapter 4

Solution Methodologies

The following sections of this chapter describe the solution procedures proposed for Ursula's pose determination problem.

4.1 Solution Procedures

The solution procedures discussed in this context have a fundamental underlying structure. The procedures start with a good initial feasible solution and, according to a fixed rule, determine a direction of movement. A move towards the point where the objective function value is at a minimum (or near minimum), is made along that direction. This is known as a line search step and may be conducted exactly or inexactly. This is repeated iteratively until a suitable termination criteria is satisfied. The primary differences between algorithms lie in the manner in which the search directions are generated. Each iteration normally involves evaluating the objective function and its derivatives at the current point, and evaluating the objective function at trial steps along the generated search directions. The intent is to obtain the final solution with as few function evaluations as possible.

4.1.1 One Dimensional Search Algorithms

The one-dimensional search algorithms form the backbone of many algorithms for solving a nonlinear programming problem [Bazaraa, Sherali and Shetty (1993)]. For a given point x_k and a direction vector d_k, determining a new point x_{k+1} involves finding a suitable step size λ_k that minimizes $f(x_k+\lambda_k d_k)$. This is a one dimensional minimization problem in the variable λ_k. Therefore, the one dimensional search algorithms play a pivotal role in the iterative solution procedures. In an attempt to make the problem more tractable, the optimal λ is assumed to lie in an interval [a, b] called the interval of uncertainty. During the line search procedure, an attempt is made to reduce the size of the interval. The function is assumed to be error free and unimodal in this interval.

Golden Section method

The Golden Section method is a unidimensional search algorithm which is a numerical approximation of the Fibonacci method, the theoretically most efficient algorithm. This is an exact search algorithm. This method reduces the initial interval $[a_1, b_1]$ by a factor of 0.618, at every iteration. The reciprocal of this factor is called the Golden Section ratio. The main advantage of this method over the Fibonacci's scheme is that the total number of iterations need not be fixed in advance. In the next paragraph, a line search technique which can be modified to conduct inexact line searches is discussed.

Quadratic Interpolation technique

The quadratic interpolation technique or the quadratic fit method is a curve fitting procedure where a smooth curve is constructed through previously measured points in order to determine an estimate of the minimum point. This method does not require any derivative information. The procedure starts at an initial point, reads the algorithmic parameters, and achieves a Three Point Pattern (TPP). A Three Point Pattern (TPP) [Bazaraa, Sherali, and Shetty (1993), Luenberger (1984)] is a set of three points such that for step-lengths $0\le\lambda_1 <\lambda_2 <\lambda_3$, the corresponding function values satisfy $f_1 \ge f_2 \le f_3$. An initial step-length is taken to determine two of these points and a search is conducted to identify the third point, revising the first two points in the process, if necessary [Bazaraa, Sherali and Shetty (1993)]. Once the TPP is achieved, a quadratic function is constructed through these points. This function is minimized to identify a new step-length. A sketch of quadratic fit search is shown in Figure 4.1.

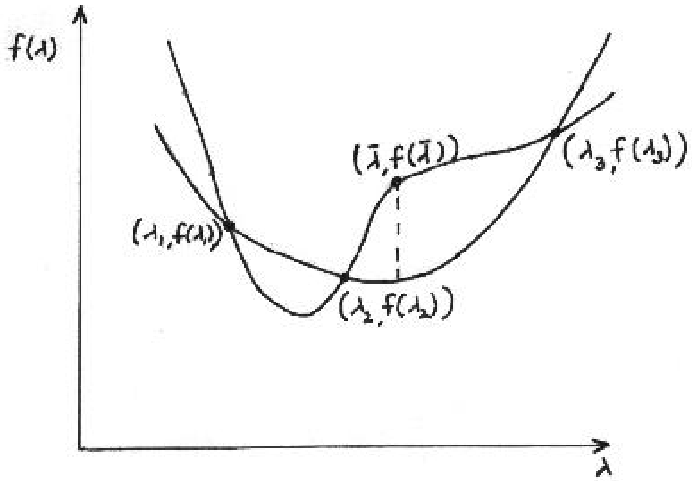

$f(\lambda)$
$(\bar{\lambda}, f(\bar{\lambda}))$
$(\lambda_3, f(\lambda_3))$
$(\lambda_1, f(\lambda_1))$
$(\lambda_2, f(\lambda_2))$
λ

Figure 4.1: Quadratic fit line search

The function value at this newly identified step-length is compared with the existing TPP, to obtain a revised three point pattern in a manner that reduces the interval of uncertainty. This process is repeated until the interval of uncertainty is sufficiently small. This technique requires one functional evaluation per iteration of the TPP search, and one per iteration of quadratic interpolation. This is an exact line search technique which can be easily modified into an inexact version.

Sometimes, due to expensive function evaluations, an exact line search may be replaced by an inexact line search technique. A line search technique which guarantees a sufficient degree of accuracy or descent in the function value would be acceptable in this case. The inexact version of the quadratic interpolation technique is similar to the exact version except that it terminates after an initial quadratic interpolation. If the quadratic approximation of the objective function is accurate enough, the first interpolation gives a good estimate of the optimal step-length. Thus, further functional evaluations to reduce the interval of uncertainty can be avoided. In the present instance of the problem, the inexact line search technique is used with quasi-Newton updates, described later in this chapter.

4.1.2 Multidimensional Search Algorithms

The multidimensional direct search algorithms are most suitable for simple constrained problems involving a relatively small number of variables. These methods converge to stationary points for differentiable functions. The search techniques require only objective function evaluations and do not use partial derivatives. In addition, information accumulated in earlier iterations may be used by some search techniques. Accordingly, given a vector *x*, a suitable direction *d* is first determined, and then the function *f* is minimized from *x* in the direction *d* by one of the one dimensional search techniques.

Hooke and Jeeves' Method

The Hooke and Jeeves' method is a derivative-free multidimensional search algorithm [Bazaraa, Sherali and Shetty (1993)]. The algorithm is an approximate steepest descent technique. The original method of Hooke and Jeeves is modified in the present context by replacing the discrete step strategy with the line search technique. This method performs two types of search - exploratory search and pattern search.

The exploratory search generates a set of orthogonal directions whose cardinality equals the number of independent variables in the problem. Line searches are made along these $d_1,.....,d_n$ coordinate directions sequentially, i.e., the new point obtained by searching in a direction d_k is taken as the initial point for the line search along the direction d_{k+1}. Search can also be made along the negative extensions of these directions.

The pattern search direction, or the acceleration direction generated by the pattern search, is employed after the exploratory searches have been conducted. This direction is determined by joining the initial point of the current iteration to the final point obtained after conducting the exploratory searches. Once the direction is determined, a line search is performed along this direction and the algorithm moves to a new iterate. The search can be terminated based on suitable criteria such as the improvement in the objective function value over one to three consecutive iterations.

The following class of solution procedures require the evaluation of gradients. These algorithms are robust and are well suited for minimizing both quadratic and nonquadratic functions.

4.1.3 Conjugate Gradient Methods

Conjugate gradient methods are derivative based gradient deflection methods that deflect the negative gradient direction using an additive deflection parameter [Bazaraa, Sherali and Shetty, (1993)]. The deflection parameter derivation is based on the function evaluation and gradient evaluation at the current iterate. These methods were proposed by Hestenes and Stiefel in 1952 for solving a linear system of equations. The methods were later modified by Fletcher and Reeves in 1964 to solve general unconstrained minimization problems.

Conjugate gradient methods can be considered as an intermediary between the steepest descent method and the quasi-Newton methods. They tend to have a faster convergence rate than the steepest descent method while avoiding the storage requirements associated with the quasi-Newton methods. If the objective function is quadratic, then, by searching along the conjugate directions, in any order, the local minimum can be found in at most n iterations, where n is the number of independent variables in the problem. The simplicity of the formula used for the generation of new direction, is an added advantage of this method. Accordingly, the algorithm generates a sequence of iterates given by,

$$y_{j+1} = y_j + \lambda_j d_j \qquad (4.1)$$

where d_j is the search direction and λ_j is the step-length which minimizes the objective function along d_j from the point y_j. For *j*=1, the search direction $d_1 = -\nabla f(y_1)$ is used, and for subsequent iterations, the search direction is determined via,

$$d_{j+1} = -\nabla f(y_{j+1}) + \alpha_j d_j \qquad (4.2)$$

where α_j is a suitable deflection parameter. In the Fletcher and Reeves' method, this is given by,

$$a_j = \frac{\|\nabla f(y_{j+1})\|^2}{\|\nabla f(y_j)\|^2}. \qquad (4.3)$$

An alternate choice based on a modified version of the conjugate gradient method, which employs the quasi-Newton condition in determining the deflection parameter, and is especially designed to be used under inexact line searches [Sherali and Ulular (1990)] is given by,

$$a_j = \frac{\left[\nabla f_{j+1} - \nabla f\right]^t \left[\nabla f_{j+1}\right] - \left(1/l_j\right)\left[y_{j+1} - y_j\right]^t \left[\nabla f_{j+1}\right]}{\left[\nabla f_{j+1} - \nabla f\right]^t \left[d_j\right]} \qquad (4.4)$$

where y_{j+1} and y_j are present and previous iterates respectively. This choice of the deflection parameter recognizes the inaccuracies in the line searches.

A pure steepest descent step is taken every n steps (n is number of independent variables in the problem). This is an important aspect that induces global convergence. It has been also empirically observed that the algorithm performs better when it is restarted after n iterations or when an insufficiently negative directional derivative is encountered or when the orthogonality between successive gradients is significantly lost [Bazaraa, Sherali, and Shetty, (1993)]. The choice of the parameter α in Equation (4.2) is also a crucial factor in the performance of the algorithm.

4.1.4 Quasi-Newton Methods

Quasi-Newton methods are conjugate direction based methods in which the gradient direction is deflected using multiplicative deflection strategies. The deflection is performed by the premultiplication of a positive definite symmetric matrix that approximates the inverse of the

Hessian matrix, with the negative gradient of the objective function [Bazaraa, Sherali and Shetty (1993)].

The descent directions generated by the quasi-Newton methods are mutually conjugate (in the quadratic case) with respect to the positive definite Hessian matrix. If the function is quadratic, it would be minimized in at most n steps, where n is the number of independent variables in the problem. The positive definite approximation matrices converge to the (inverse of the) Hessian matrix of the function when the minimum is approached. Updating this positive definite symmetric matrix involves adding two symmetric matrices, each of rank one, using the popular BFGS method. This procedure is therefore classified as a rank-two correction procedure. Using this correction procedure, the quasi-Newton procedures overcome the problems with Newton method, namely, the possibility of the Hessian matrix being singular and the absence of positive definiteness.

The path pursued by the quasi-Newton algorithm is similar to the conjugate gradient method, and usually enjoys faster convergence. For nonquadratic functions, the idea of restarting the algorithm after every n iterations (n is number of independent variables in the problem), is theoretically prescribed to induce convergence, but is sometimes ignored in empirical implementations. The following paragraph discusses redesigning the above solution procedures to solve the problem in the present context.

4.2 Redesigning the Solution Procedures

The laser gun calibration problem involves certain nonnegativity constraints. Thus, the solution procedures discussed above need to be redesigned to incorporate these constraints. Barrier functions may be used to transform a constrained problem into an unconstrained problem [Bazaraa, Sherali, and Shetty (1993)]. These functions set a barrier against leaving the feasible region. If the optimal solution occurs at the boundary of the feasible region, the algorithm moves from the interior to the boundary.

The solution procedures also need to be redesigned to evaluate the gradient at a given point. The behavior of the objective function being largely unknown, and the difficulty involved in finding the derivatives analytically, encourage the use of finite differences to approximate the gradient at any given point. Accordingly, the gradient of the objective function at the point *x* may be computed using the forward differences method. Let x_{j} be the *j*th component of the point *x*, let ***d*** be a parameter, and let x_{j} be the *j*th component of the gradient being evaluated at the point *x*. According to the forward differences method,

$$\xi_j = \frac{f(x_j + \delta) - f(x_j)}{\delta} \forall j = 1,....,n. \tag{4.5}$$

An accurate gradient evaluation would result in better search directions and hence reduce the number of iterations. The choice of the parameter δ is crucial in this regard. A relatively large δ contains information regarding a considerable local neighborhood and may not be a good estimate of the gradient at that point. Directions generated using this gradient would result in the descent to the vicinity of an optimum faster, but experience convergence difficulties near optimum. A relatively smaller δ provides a good local estimate of the gradient and hence, is good for the final convergence purpose. Therefore, the choice of the parameter δ is important. In our computations, we used $\delta = 0.0001$. This was based on the results obtained from the test problems.

4.3 Test Problems

The following problems are used to test the validity of the model and to compare the computational competitiveness, numerical stability and accuracy of the proposed solution procedures. The ideal case scenario in the laser gun pose determination problem is used to check the sensitivity of the model and its ability to solve a real-life problem. Similarly, the Ursula pose determination problem can be tested using the ideal case scenario. A collection of test problems [Sherali and Ulular, (1990)], chosen from the standard test problems used in the literature, was used to study the performance of various nonlinear search procedures. These test problems have a variety of objective function structures. The performance of the search procedures when applied to such test problems, would provide information on the performance of the search procedures when used to solve highly nonlinear problems.

4.3.1 Testing the Model for the Laser Gun Pose Determination Pr oblem

An ideal case scenario, in which it is assumed that no kinematic errors are present in the setup, is used to test the model developed to solve the laser gun pose determination problem. Accordingly, the targets are placed at locations given by the coordinates [80.0, 0, 0], [0,80.0, 0], [-80.0,0,0] and [0,-80.0,0]. The model is then used to calculate the pan and tilt angles made by the laser gun to shoot the targets placed at locations mentioned earlier. Hence, the pose parameters of the laser gun can be found using the solution procedures described earlier.

4.3.2 Testing the Model for the Ursula Location Determination Problem

An ideal case scenario, similar to the laser gun pose determination problem, can be used to test the model developed for determining Ursula's location. Accordingly, Ursula may be assumed to be situated at an arbitarily chosen location. Based on its location, the theoretical pan and tilt angles made by the laser gun to shoot these targets can be calculated analytically. When these parameters are fed, the model should return the chosen location of Ursula. Thus, the validity of the model can be tested.

4.3.3 Standard Test Problems for the Search Procedures

The following functions [Sherali and Ulular, (1990)] have been used to test the solution procedures.

1. Witte and Holst’s Strait function

Starting solution: (-1,2, 1.0)

Final objective function value : 0

$$\left(\left(x_2 - x_1^2\right)^2 + 100(1 - x_1)^2\right) \tag{4.6}$$

2. Witte and Holst’s Cube function:

Starting solution: (-1.2, 1.0)

Final objective function value: 0

$$100\left(x_2 - x_1^3\right)^2 + \left(1 - x_1\right)^2 \tag{4.7}$$

3. C.F. Wood’s Function:

Starting Solution: (-3,-1,-3,-1)

Final objective function value : 0

$$100\left(x_2 - x_1^2\right)^2 + \left(1 - x_1\right)^2 + 90\left(x_4 - x_3^2\right)^2 + \left(1 - x_3\right)^2 + 10.1\left(x_2 - 1\right)^2 + 10.1\left(x_4 - 1\right)^2 + 19.8\left(x_2 - 1\right)\left(x_4 - 1\right) \tag{4.8}$$

4. Powell’s function:

Starting solution: (-3,-1,0,1)

Final objective function value: 0

$$\left(x_1 + 10x_2\right)^2 + 5\left(x_3 - x_4\right)^2 + \left(x_2 - 2x_3\right)^4 + 10\left(x_1 - x_4\right)^4 \tag{4.9}$$

5. Witte and Holst’s Shallow function:

Starting solution: (-3, -1.0)

Final objective function value: 0

$$\left(x_2 - x_1{}^2\right)^2 + \left(1 - x_1\right)^2 \qquad (4.10)$$

Chapter 5

Implementation Guidelines and Results

This chapter documents the experiments conducted using the solution procedures on the test problems described in the previous chapter. The implementation of the solution procedures and the modifications made to them are discussed in this context. The solution procedures are described earlier in Chapter 4. Comparisons among the solution procedures are drawn based on their computational competitiveness, convergence properties, numerical stability and the simplicity of implementation.

5.1 Solution procedures

The solution procedures which were used to solve the laser gun pose determination and Ursula's location determination problems are

1. Hooke and Jeeves method,
2. Conjugate gradient method using exact line searches (Fletcher and Reeves' method),
3. Conjugate gradient method using inexact line searches (Sherali and Ulular), and
4. Quasi-Newton method using BFGS updating scheme.

The Golden Section method was used as the line search technique for implementing the Hooke and Jeeves, Fletcher and Reeves, and the quasi-Newton methods. The quadratic interpolation technique was used for implementing the conjugate gradient method with inexact line searches. The final interval of uncertainty for the Golden Section technique was fixed at 0.001. The initial step-length for the quadratic interpolation technique was fixed at 5.0. The termination criteria for the Hooke and Jeeves method was based on the present and previous iterates given by x_k and x_{k-1} respectively. If $||x_k - x_{k-1}|| \leq \xi$, where ξ is a scalar fixed at a value of 0.0001, then the algorithm was terminated. For the other methods, the termination criteria was based on the gradient at the current iterate x_k, given by $\nabla f(x_k)$. If $||\nabla f(x_k)|| \leq \xi$, then the algorithms were terminated. The gradient evaluations required for these methods were done using the finite differences technique discussed in the earlier chapter. The parameter δ was fixed at 0.0001. The solution procedures and the line search techniques have been coded in C++ programming language and their executable versions were run on a 133 mHz Pentium PC. The solution procedures were tested using the problems listed in Section 4.3.3. The results of these test problems are tabulated below.

Table 5.1: Results of the Hooke and Jeeves Method

Pr #	*n*	Final solution	# of fn evals	# of iters	Initial obj fn. value	Final obj fn. value	Time in secs
1	2	(1.0,0.99)	126	1	484.19	8.42×10^{-7}	≈0
2	2	(0.99,0.99)	124	1	749.08	0.0002	≈0
3	4	(1.25, 1.58, 0.62, 0.38)	224	1	19192	0.359	1.1
4	4	(-0.23,0.02,-0.04,-0.06)	496	1	2735	0.01	≈0
5	2	(1.0, 1.0)	247	2	116	8.57×10^{-8}	≈0

Table 5.2: Results of the Fletcher and Reeves Method							
Pr #	*n*	Final solution	#. of fn evals	#.of iters	Initial obj. fn value.	Final obj fn.value	Time in secs
1	2	(1.0,1.0)	129	3	484.19	1.8x10^{-5}	≈0
2	2	(0.99,0.97)	1446	15	749.08	8.99x10^{-5}	≈0
3	4	(0.97, 0.99, 1.0, 1.0)	2284	534	19192	1.25x10^{-6}	0.6
4	4	(-0.07,0.007,-0.041,-0.041)	2582	11	2735	7.43x10^{-5}	0.05
5	2	(0.99, 0.99)	308	4	116	1.13x10^{-5}	≈0

Table 5.3: Results of the Sherali and Ulular's Method							
Pr #	*n*	Final solution	# of func evals	# of iters	Initial obj. fn value	Final obj fn. value	Time in secs
1	2	(0.99,0.99)	615	8	484.19	2.14x10^{-5}	≈0
2	2	(0.98,0.95)	94458	936	749.08	2.0x10^{-4}	0.55
3	4	(0.99, 0.99, 1.0, 1.0)	383	6	19192	2.29x10^{-4}	≈0
4	4	(-0.08,0.008,-0.05, -0.05)	1946	9	2735	1.0x10^{-4}	0.06
5	2	(0.99, 0.99)	892	9	116	2.3x10^{-6}	≈0

Table 5.4: Results of the Quasi-Newton Method with BFGS Updates							
Pr #	*n*	Final solution	# of func evals	# of iters	Initial obj. fn value	Final obj fn. value	Time in secs
1	2	(0.99,0.99)	180	2	484.19	5.79x10^{-8}	≈0
2	2	(0.98,0.95)	6833	63	749.08	1.02x10^{-5}	0.05
3	4	(0.99, 0.99, 1.0, 1.0)	42809	152	19192	1.31x10^{-7}	0.5
4	4	(-0.08,0.008,-0.05,-0.05)	1023	4	2735	1.56x10^{-5}	0.06
5	2	(0.99, 0.99)	290	3	116	1.05x10^{-5}	≈0

The first column in tables 5.1, 5.2, 5.3 and 5.4 represents the test problem number. The number of variables are mentioned in the second column. The final solution to the problem is stated in the third column. The number of function evaluations, number of iterations, initial and final objective function values are mentioned in the fourth, fifth, sixth and seventh columns respectively. The CPU time required for the execution of the algorithm is given in the last column.

The results of the test problems show that the solution procedures converge to the near optimal solutions in a reasonable number of iterations. The solution procedures were next fine tuned for their application to the laser gun pose determination and Ursula's location determination problems.

5.2 Laser gun pose determination problem

The laser gun pose determination corresponds to the first phase of the problem solution. The problem mathematically modeled in (3.18), involved minimizing a least squares objective function. The model developed in (3.18) can be tested using the ideal case scenario discussed in Section 4.3.1. Accordingly, the targets are placed at locations given by the coordinates [80.0, 0, 0], [0,80.0,0], [-80.0,0,0] and [0,-80.0,0]. The parameters θ_I, α_0, d_1, ϕ_I, α_1, d_2, α_L, β_L, γ_L, x_L, y_L and z_L are fixed at their nominal values, which are listed in Appendix A. The objective function is constructed by substituting these parameters in Equations (3.6), (3.7), (3.8), (3.9) and (3.10). The procedure is discussed in Section 3.2. The pose parameters {x, y, z, α, β, γ}, in the ideal case would be given by {0,0,0,0,0,0}. The initial values of the variables {α, β, γ, p_1, p_2, p_3, p_4}which correspond to the initial solution of the problem were given by {0.785, 0.785, 0.785, 45, 45, 45, 45}. The results of the test problem using the solution procedures discussed earlier, are tabulated in Table 5.5.

Table 5.5: Results of Test Problem 4.3.1 Using Different Solution Procedures

Solution procedures	Final solution	# of fn. evals	# of iters	Init. obj. fn value	Final obj fn. value	Time in secs
Hooke and Jeeves method	(0.0001,-0.0004, 0.003, 6.1x10^{-5}, -0.0001, -0.0001)	3211	2	9810.7	1.27x10^{-8}	0.33
Fletcher and Reeves method (Exact line search)	(-0.001,-0.0004 0.004, 0.0001, 9.9x10^{-5}, 0.006)	1539	4	9810.7	1.90x10^{-5}	0.22
Sherali and Ullular's method (Inexact line search)	(-6.5x10^{-5},-.0004 0.0037, 9.8x10^{-5}, -0.001, 0.0006)	1374	3	9810.7	2.02x10^{-5}	0.22
BFGS method	(-8.7x10^{-8},-.0004 0.0037, 7.2x10^{-5}, -0.0001, 1.2x10^{-7})	1359	3	9810.7	1.21x10^{-5}	0.22

The results tabulated in Table 5.5 show that the solution procedures have found a near optimal solution to the test problem discussed in Section 4.3.1, over a finite number of iterations. Each solution procedure is characterized by a unique solution to the problem. The BFGS method found the best solution among the final solutions obtained by solution procedures, in least amount of CPU time. The number of function evaluations performed by gradient based methods are far lesser than the Hooke and Jeeves method. This can be explained by the fact that the gradient based methods are characterized by faster convergence rates. The gradient based algorithms were implemented with the restart technique (discussed in Chapter 4) which resulted in a faster rate of convergence.

In order to evaluate the performance of the solution procedures in the real-life scenario, the kinematic errors have been included in the model. The procedure for modeling these errors is discussed in Section 3.3. The ranges of error are listed in Appendix A. The errors were generated randomly using the algorithm described in Law and Kelton (1991). The determination of q_i and f_i (i =1,2,3,4) values which corresponds to Step 3 of the error modeling procedure, was done using the Hooke and Jeeves method. These values were given as the input to the problem (3.18). The problem (3.18) was then solved using the solution procedures discussed in Section 5.1. Ten different problems have been generated using different random seeds. Each seed generates a unique set of the error parameters and thus corresponds to a unique real-life problem. The theoretical solution to the ideal case scenario was used as the corresponding initial solution to all the problems. The results of application of the solution procedures to these problems are tabulated and discussed in the following paragraphs.

Hooke and Jeeves method

The Hooke and Jeeves method was described in Chapter 4. Its theoretical convergence argument is provided in the text by Bazaraa, Sherali, and Shetty (1993). The implementation of the algorithm uses the positive and negative coordinate axes for performing the exploratory searches. The performance of the algorithm is sensitive to the definition of the parameter ξ, which has been fixed at 0.0001. The termination criteria for this algorithm was discussed earlier. The results of this solution procedure are tabulated in Table 5.6.

Table 5.6: Results of the Hooke and Jeeves Method

Random seed	Final solution ($x,y,z,\alpha,\beta,\gamma$)	# of func evals	# of iters	Init obj. fn value	Final obj.fn value	Time in secs
1	(-0.3, -0.34, -0.003, -1.486, -0.0002, $-3.25\text{x}10^{-5}$)	3209	2	25.75	0.0088	0.28
2	(-0.33, -1.03, -0.0003, 0.71, -0.00026, -0.00017)	3201	2	6.97	1.08	0.28
3	(-0.311, 0.225, $-1.54\text{x}10^{-5}$, 0.2, $-1.1\text{x}10^{-5}$, $-1.1\text{x}10^{-5}$)	1604	1	0.512	0.032	0.11
4	(-0.1479, 0.28, -0.0003, 1.095, -0.00026, -0.0002)	1604	1	14.02	0.0099	0.16
5	(-0.147, -1.328, -0.0003, 0.9393,-0.00026,-0.00004)	12315	8	12.38	2.07	1.1
6	(-0.13,0.199,$-5.79\text{x}10^{-6}$, -0.3811,$-4.16\text{x}10^{-6}$, $-4.16\text{x}10^{-6}$)	1604	1	1.69	$4.53\text{x}10^{-8}$	0.11
7	(-0.2319,0.14,-0.0003,-0.72, -0.00026,-0.00026)	3778	3	6.22	0.028	0.33
8	(-0.28, -1.25, -0.0003, 0.904, -0.0002, -0.0004)	4812	3	11.36	1.80	0.38
9	(-0.2538,0.099,-0.0003,-0.072, -0.0001,-0.0001)	11224	7	0.104	0.043	0.99
10	(-0.199,0.285, -0.0003, -0.89, -0.00026, -0.00026)	1606	1	9.254	0.0019	0.17

The results show that the function evaluations in the real-life scenario are costlier than the ideal case. The algorithm converges to near optimal solutions in all the test problems except for the 2nd, 5th and 8th test problems. The effect of decreasing the parameter ξ further, did not change the final solution of the problems but sometimes increased the number of iterations and function evaluations. The final objective function values and the corresponding solutions reinforce the convergence properties of the algorithm for the different test problems. The number of function evaluations and the number of iterations are also recorded in the table for later comparisons.

Conjugate Gradient Method (Fletcher and Reeves Method)

The theoretical motivation behind this algorithm has been discussed in Chapter 4. The performance of this algorithm is sensitive to the definition of the initial interval of uncertainty, required for conducting the line search. This interval was fixed at [0, 10]. The evaluations of the gradient were done using the finite differences technique discussed in Chapter 4. Extremely low values of the parameter δ (like 0.0000001) cause numerical instability. Care was taken that to avoid divisions by numerically insignificant denominators while evaluating the Euclidean norms of the gradients and the gradients themselves. Restarting the algorithm after n (n is the number of independent variables in the problem) iterations, proved to be effective. The results are tabulated in Table 5.7.

Table 5.7: Results of the Fletcher and Reeves Method

Random seed	Final solution (x,y,z,α,β,γ)	# of func evals	# of iters	Init obj. fn value	Final obj.fn value	Time in secs
1	(-0.30, -0.35, 0, -1.486, 0, 0)	3132	6	25.75	0.0088	0.44
2	(-0.33, -1.03, 0, 0.71, 0, 0)	791	3	6.97	1.08	0.22
3	(-0.311, 0.225, 0, -0.2, 0, 0)	625	2	0.512	0.032	0.22
4	(-0.111, 0.28, 0, 1.095, 0 ,0)	625	2	14.02	0.0099	0.22
5	(-0.147, -1.328, -0.0003 0.9393, 0 ,0)	694	3	12.38	2.07	0.22

6	(-0.13,0.199, 0, -0.3811, 0, 0)	625	2	1.69	1.41x10^{-8}	0.22
7	(-0.2319,0.14, 0, -0.72, 0, 0)	694	3	6.22	0.028	0.28
8	(-0.28, -1.25, 0, 0.904, 0, 0)	625	2	11.36	1.80	0.17
9	(-0.2538,0.098, 0, -0.072, 0, 0)	625	2	0.104	0.043	0.22
10	(-0.2,0.285, 0, -0.89, 0, 0)	694	3	9.254	0.0019	0.22

Table 5.7 records the computational experience with the conjugate gradient method using exact line searches. The algorithm exhibits a faster convergence performance than Hooke and Jeeves's method. Increasing the parameter ξ by a factor of 10, did not result in significant changes in the final solutions. However, decreasing this parameter by a factor of 10, resulted in a significant increase in the number of function evaluations and iterations. This method will be compared with an alternate version that uses inexact line searches later in this chapter.

Sherali and Ulular's Method (Conjugate gradient method using Inexact Line Searches)

The implementation of this procedure is similar to the Fletcher and Reeves method. The initial interval of uncertainty is not required in this case. The procedure conducts the line searches using the quadratic interpolation technique, which utilizes the shape of the objective function. The method of updating the search directions is discussed in Chapter 4. The results using this solution procedure are tabulated in Table 5.8.

Table 5.8: Results of the Sherali and Ulular’s Method

Random seed	Final solution $(x,y,z,\alpha,\beta,\gamma)$	# of func evals	# of iters	Init obj. fn value	Final obj.fn value	Time in secs
1	(-0.29, -0.35, 0, -1.486, 0, 0)	534	2	25.75	0.0088	0.06
2	(-0.33, -1.03, 0, 0.71, 0, 0)	688	2	6.97	1.08	0.06
3	(-0.311, 0.225, 0, -0.2, 0, 0)	691	2	0.512	0.032	0.05
4	(-0.111, 0.28, 0, 1.095, 0 ,0)	703	2	14.02	0.0099	0.06
5	(-0.14, -1.32, -0.0003, 0.939, 0,0)	626	2	12.38	2.07	0.06
6	(-0.13,0.199, 0, -0.3811, 0, 0)	540	2	1.69	1.4x10^{-8}	0.05
7	(-0.2319,0.14, 0, -0.72, 0, 0)	488	1	6.22	0.028	0.06
8	(-0.28, -1.25, 0, 0.904, 0, 0)	534	2	11.36	1.80	0.05
9	(-0.2538,0.098, 0, -0.072, 0, 0)	482	1	0.104	0.043	0.06
10	(-0.2,0.285, 0, -0.89, 0, 0)	488	1	9.254	0.0019	0.04

The inexact line search version of the conjugate gradient method, gave the final solution in a remarkably low CPU time. The number of function evaluations and iterations are also recorded and will be addressed later for drawing comparisons among the solution procedures.

Quasi-Newton method

The implementation of the quasi-Newton method is identical to the implementation of the conjugate gradient methods except for the direction generation technique. The direction generation procedure involves updating a deflection matrix (approximation to the inverse Hessian), ensuring the positive definiteness of this matrix, and deflecting the negative gradient at the current iterate by the multiplying with this matrix. The direction generation using the BFGS updating scheme is described in text by Bazaara, Sherali and Shetty (1993).

The solution procedure employs the Golden Section exact line search technique. The parameters discussed for the case of the conjugate gradient methods were also used for this solution procedure. The results are tabulated in Table 5.9.

Table 5.9: Results of the Quasi-Newton Method Using BFGS Updates						
Random seed	Final solution (x,y,z,α,β,γ)	# of func evals	# of iters	Init obj. fn value	Final obj.fn value	Time in secs
1	(-0.3, -0.35, 0, -1.486, 0, 0)	1189	2	25.75	0.0088	0.22
2	(-0.33, -1.03, 0, 0.71, 0, 0)	623	1	6.97	1.08	0.17
3	(-0.311, 0.225, 0, -0.2, 0, 0)	425	1	0.512	0.032	0.22
4	(-0.113, 0.28, 0, 1.095, 0 ,0)	425	1	14.02	0.0099	0.22
5	(-0.14, -1.32, -0.0003,0.93,0,0)	623	1	12.38	2.07	0.22
6	(-0.13,0.199, 0, -0.3811, 0, 0)	524	1	1.69	1.41×10^{-8}	0.22
7	(-0.2319,0.14, 0, -0.72, 0, 0)	425	1	6.22	0.028	0.22
8	(-0.28, -1.25, 0, 0.904, 0, 0)	621	1	11.36	1.80	0.22
9	(-0.2538,0.098, 0, -0.072, 0, 0)	425	1	0.104	0.043	0.22
10	(-0.2,0.285, 0, -0.89, 0, 0)	425	1	9.254	0.0019	0.22

The results show that the Quasi-Newton method converged to near optimal solution in atmost 2 iterations. This is an indication of the high rates of convergence exhibited by these methods. The BFGS method consistently obtains the final solution with less number of iterations and function evaluations. The overall comparisons between the methods are drawn in the following paragraph.

Comparisons between the solution procedures

The final solutions obtained by the different solution procedures for the randomly generated problems, are very similar. The inexact line search version of the Sherali and Ulular's conjugate gradient method and the BFGS quasi-Newton method perform better than the rest of the solution procedures. The former takes significantly less time to obtain the final solution. In the case of the first test problem, the conjugate gradient method (inexact line search) outscores all the solution procedures. It also appears to be more robust than the Fletcher and Reeves method.

5.3 Ursula's location determination problem

The determination of Ursula's location corresponds to the second phase of the problem solution. This involves the solution of a least squares minimization problem modeled in (3.21). The input parameters for the problem are the laser gun pose, rotation encoder readings (pan and tilt angles) and the depth of Ursula. The radius of the reactor vessel was fixed at 90 units. The determination of the laser gun pose is discussed in Section 5.2. The readings from the rotation encoders located on the pan and tilt axes, may be obtained when the photosensor target located on Ursula is shot. The depth at which Ursula is located, is obtained from a depth gauge. The solution procedures discussed in Section 5.1 were used to solve the problem (3.21). The implementation of the solution procedures for this problem is identical to the implementation for the laser gun pose determination problem. Two additional solution procedures which are based on the alternate approach discussed in (3.22), were used to solve the problem. They are

1. Golden Section method
2. Quadratic interpolation or the quadratic fit method.

The models developed in (3.21) and (3.22) were tested using the test problem discussed in Section 4.3.2. Accordingly, Ursula was assumed to be located at [90, 0]. The depth at which Ursula is located was chosen as 90 units. The theoretical pan and tilt angles for this position, can be calculated analytically and are given by 0 and 45 degrees respectively. The laser gun pose was given by the theoretical solution to the test case discussed in Section 4.3.1, i.e., {*x*, *y*, *z*, *a*, *b*, *g*} were fixed at {0, 0, 0, 0, 0, 0}. The laser gun pose along with the depth parameter and the rotation encoder readings, were given as input to the model. The results of the solution procedures when applied to this test problem are tabulated in Table 5.10.

Table 5.10: Results of the Test Problem 4.3.2 Using Different Solution Procedures

Solution Procedure	Final solution (x, y)	# of fn. evals	# of iters	Init. obj fn value	Final obj fn. value	Time in secs
Hooke and Jeeves method	(90, 0.009)	1678	11	5214.1	4.1x10^{-7}	0.38
Fletcher and Reeves method	(90, 0.009)	1681	12	5214.1	4.2x10^{-7}	0.38
Conj. grad (Inexact)	(90, 0.008)	180	4	5214.1	4.5x10^{-7}	0.33
BFGS method	(90, 0.008)	1667	14	5214.1	9.4x10^{-8}	0.33
Golden Section method	(90, 0.004)	52	48*	4392.6	9.2x10^{-6}	≈0
Quadratic fit method	(90, 0.008)	46	3*	4392.6	3.8x10^{-11}	≈0

* These are the univariate line search iterations.

The results of the solution procedures validate the model. The final solution obtained by these procedures confirms with the arbitrarily chosen location. The solution procedures which are based on the alternate formulation (3.22) performed better than the others. The conjugate gradient algorithm (Sherali and Ulular's method) with inexact line searches converges to the near optimal solution in a significantly less number of iterations and function evaluations. The Fletcher and Reeves algorithm was modified by normalizing the initial direction which is given by the negative gradient direction at the starting iterate. This reduced the number of iterations and the execution time to a considerable extent.

The solution procedures were tested in real-life scenario by including the kinematic errors in the model. This procedure is discussed in Section 3.3. Accordingly, two different laser gun poses which were obtained by taking the kinematic errors into account, were used. These correspond to the two problem sets. The pose parameters $\{x, y, z, a, b, g\}$ for the problem sets are given by {-0.2538, -0.099, -0.000, -0.0720, -0.0001, -0.0001}and {-0.1479, -1.328, -0.0003, -0.9393, -0.00026, -0.0004}respectively. These poses were obtained from the results in Section 5.2 (solutions to problems # 9 and 5 respectively). Five different problems have been generated in each set using different random seeds and which correspond to different locations of the Ursula. The results of solution procedures are recorded in the tables 5.11, 5.12, 5.13, 5.14, 5.15 and 5.16. The objective function contour of a sample test problem (the 5th problem of the first set) is attached in Appendix B.

Table 5.11: Results of the Hooke and Jeeves Method

Random seed	Input parameters (Pan angle, Tilt angle, Depth)	Final solution (x, y)	# of fn. evals	# of iters.	Init obj. fn value	Final obj fn value	Time in secs
	Problem set 1						
1	(0, 45, -90)	(-89.3, -6.38)	3669	24	5700.46	0.46	0.44
2	(90,45,-90)	(6.79, 89.97)	1841	12	4697.56	0.065	0.38
3	(180,45,-90)	(-89.68,6.59)	927	6	18957.6	0.214	0.43
4	(270,45,-90)	(-6.21, -89.4)	925	6	19962.7	0.057	0.44
5	(90,0,0)	(6.78, 90.07)	768	5	3604	0.031	0.39
	Problem set 2						
6	(0,45,-90)	(53.82, -72.16)	611	4	13664	0.955	0.44
7	(90,45,-90)	(71.86, 53.71)	460	3	2261	0.67	0.39
8	(180,45,-90)	(-52.6, 73.5)	1666	8	10841.6	0.056	0.38
9	(270,45,-90)	(-73.32, -52.39)	450	3	22203.1	0.926	0.38
10	(90,0,0)	(71.92, 53.78)	458	3	225.03	0.003	0.44

The Hooke and Jeeves method performs reasonably well and obtains the final solution making good descent toward near optimal solutions. The problem set 2 requires significantly fewer iterations and function evaluations than the problem set 1. The Hooke and Jeeves algorithm appears to be robust and the results did not change with a decrease in the parameter ξ. The comparisons among different solution procedures are drawn later in this chapter.

Table 5.12: Results of the Fletcher and Reeves Method

Random seed	Input parameters (Pan angle, Tilt angle, Depth)	Final solution (x, y)	# of fn evals	# of iters.	Init obj. fn value	Final obj fn value	Time in secs
	Problem set 1						
1	(0, 45, -90)	(-89.3, -6.38)	15957	131	5700.46	0.46	0.55
2	(90,45,-90)	(6.79, 89.97)	72713	591	4697.56	0.065	1.09

3	(180,45,-90)	(-89.68,6.59)	89305	725	18957.6	0.214	1.26
4	(270,45,-90)	(-6.21, -89.4)	23317	191	19962.7	0.057	0.61
5	(90,0,0_	(6.78, 90.07)	71915	584	3604	0.031	1.1
	Problem set 2						
6	(0,45,-90)	(53.82, -72.16)	106695	865	13664	0.955	1.38
7	(90,45,-90)	(71.86, 53.71)	52521	427	2261	0.67	0.93
8	(180,45,-90)	(-52.6, 73.5)	42787	348	10841.6	0.056	0.77
9	(270,45,-90)	(-73.32,-52.39)	71855	583	22203.1	0.926	1.1
10	(90,0,0)	(71.92, 53.78)	72132	585	225.03	0.003	1.1

The Fletcher and Reeves method was modified by normalizing the initial direction which was given by the negative gradient at the starting iterate. This modification improved the convergence properties of the algorithm. In spite of the modification, it can be observed that the algorithm takes considerable time in finding a near optimal solution for the second set of problems. The algorithm was found to be sensitive to the parameter ξ. This was increased from 0.0001 to 0.01 to induce faster convergence. The algorithm was also found to be sensitive to the definition of the initial interval of uncertainty for the line search procedure. The results of the inexact line search version are recorded in Table 5.13.

Table 5.13: Results of the Sherali and Ulular’s Method

Random seed	Input parameters (Pan angle, Tilt angle, Depth)	Final solution (x, y)	# of fn. evals	# of iters.	Init obj. fn value	Final obj fn value	Time in secs
	Problem set 1						
1	(0, 45, -90)	(-89.3, -6.38)	598	7	5700.46	0.46	0.44
2	(90,45,-90)	(6.79, 89.97)	186	4	4697.56	0.065	0.38
3	(180,45,-90)	(-89.68,6.59)	658	9	18957.6	0.214	0.39
4	(270,45,-90)	(-6.21, -89.4)	2143	26	19962.7	0.057	0.44

5	(90,0,0)	(6.78, 90.07)	14920	136	3604	0.031	0.54
	Problem set 2						
6	(0,45,-90)	(53.82, -72.16)	11874	112	13664	0.955	0.49
7	(90,45,-90)	(71.86, 53.71)	10530	97	2261	0.67	0.5
8	(180,45,-90)	(-52.6, 73.5)	3862	37	10841.6	0.056	0.44
9	(270,45,-90)	(-73.32, -2.39)	14191	130	22203.1	0.926	0.49
10	(90,0,0)	(71.92, 53.78)	7519	67	225.03	0.003	0.49

The inexact line search version of the conjugate gradient method (Sherali and Ulular's method) performs better than the Fletcher and Reeves method in every aspect. The problems in the second set require more iterations than the first set. The restart technique proved to be very effective in this case too. The results of the quasi-Newton method are tabulated in Table 5.14.

Table 5.14: Results of the Quasi-Newton Method with BFGS Updates

Random seed	Input parameters (Pan angle, Tilt angle, Depth)	Final solution (x, y)	# of fn. evals	# of iters.	Init obj. fn value	Final obj fn value	Time in secs
	Problem set 1						
1	(0, 45, -90)	(-89.3, -6.38)	1287	11	5700.46	0.46	0.38
2	(90,45,-90)	(6.79, 89.97)	2114	17	4697.56	0.065	0.44
3	(180,45,-90)	(-89.68,6.59)	4473	36	18957.6	0.214	0.44
4	(270,45,-90)	(-6.21, -89.4)	1600	13	19962.7	0.057	0.44
5	(90,0,0)	(6.78, 90.07)	4573	37	3604	0.031	0.44
	Problem set 2						
6	(0,45,-90)	(53.82, -72.16)	1350	11	13664	0.955	0.44
7	(90,45,-90)	(71.86, 53.71)	844	7	2261	0.67	0.44
8	(180,45,-90)	(-52.6, 73.5)	1858	15	10841.6	0.056	0.44
9	(270,45,-90)	(-73.3, -52.39)	2805	23	22203.1	0.926	0.39
10	(90,0,0)	(71.92, 53.78)	777	7	225.03	0.003	0.39

The results of the BFGS method clearly demonstrate the superior convergence properties of the quasi-Newton techniques. The algorithm is robust and performs well against both problem sets. The results of the solution procedures which were based on the alternate formulation (3.22) of the problem are discussed in tables 5.15 and 5.16.

Table 5.15: Results of the Golden Section Method (Based on Alternate Approach)

Random seed	Input parameters (Pan angle, Tilt angle, Depth)	Final solution (x, y)	# of fn. evals	Initial obj. fn value	Final obj. fn value	Time in secs
	Problem set 1					
1	(0, 45, -90)	(-89.3, -6.38)	50	5078.07	0.46	≈0
2	(90,45,-90)	(6.79, 89.97)	52	3690.17	0.065	≈0
3	(180,45,-90)	(-89.68,6.59)	50	15595.5	0.214	≈0
4	(270,45,-90)	(-6.21, -89.4)	50	15856.7	0.057	≈0
5	(90,0,0)	(6.78, 90.07)	48	3464.83	0.031	≈0
	Problem set 2					
6	(0,45,-90)	(53.82, -72.16)	50	13006.0	0.955	≈0
7	(90,45,-90)	(71.86, 53.71)	52	167.5	0.67	≈0
8	(180,45,-90)	(-52.6, 73.5)	48	10789.5	0.056	≈0
9	(270,45,-90)	(-73.32, -52.39)	48	15974.2	0.926	≈0
10	(90,0,0)	(71.92, 53.78)	48	166.63	0.003	≈0

The alternate formulation transforms the problem into an equivalent univariate problem by using the projection method. Thus, an exact line search technique like the Golden Section method can be easily applied. The results show that the formulation was very effective in reducing the CPU time taken. The number of function evaluations have been cut down drastically using this formulation. This method is robust and numerically very stable. The final solutions obtained by this method are identical to those obtained by the other solution procedures. The results of the quadratic interpolation technique, which uses the same formulation, are given in Table 5.16.

Table 5.16: Results of the Quadratic Interpolation Technique

Random seed	Input parameters (Pan angle, Tilt angle, Depth)	Final solution (x, y)	# of fn. evals	Initial obj. fn value	Final obj. fn value	Time in secs
	Problem set 1					
1	(0, 45, -90)	(-89.3, -6.35)	64	5078.07	0.495	≈0
2	(90,45,-90)	(6.73, 89.74)	78	3690.17	0.029	≈0
3	(180,45,-90)	(-89.7,6.55)	75	15595.5	0.16	≈0
4	(270,45,-90)	(-6.19, -89.7)	91	15856.7	0.029	≈0
5	(90,0,0)	(6.71, 89.7)	94	3464.83	0.055	≈0
	Problem set 2					
6	(0,45,-90)	(53.83, -72.11)	75	13006.0	2.37	≈0
7	(90,45,-90)	(72.06, 53.90)	49	167.5	1.979	≈0
8	(180,45,-90)	(-52.44, 73.14)	161	10789.5	2.37	≈0
9	(270,45,-90)	(-73.20, -52.35)	59	15974.2	1.97	≈0
10	(90,0,0)	(72.05, 53.92)	65	166.63	3.26	≈0

The results for the quadratic interpolation technique show an inconsistent performance, the procedure gave better final solutions for the first set of problems. The termination criteria for the algorithm is stated as follows. For a set of points [λ_1, λ_2, λ_3] satisfying TPP (description provided in Section 4.1.1), if $||\lambda_1-\lambda_3|| \leq \xi$, then the algorithm may be terminated. The value of the parameter ξ was chosen as 0.001. Decreasing the value by a factor of 10, did not improve the results for the second set of problems.

Comparison between different solution procedures

The alternate formulation of the Ursula's location determination problem proved to be very effective. The Golden Section method, which is based on this approach, performed far better than the other solution procedures. But the fact that this formulation is specific to the structure and size of the problem should be kept in mind. The BFGS method performs consistently and is robust. The Hooke and Jeeves method performs better when applied to this problem rather than the laser gun pose determination problem. This may be attributed to the lesser number of variables in the problem. The conjugate gradient method (Sherali and Ulular's method) with inexact line searches is also computationally competitive, as it is evident from the results of the first set of problems.

The results of the experiments conducted with the solution procedures on the laser gun pose determination and the Ursula location determination problems were discussed in this chapter. Conclusions and some future research opportunities are discussed in the next chapter.

Chapter 6

Conclusions and Recommendations for Future Research

This chapter discusses the conclusions derived from the present research effort and mentions some opportunities for future research.

The modeling of the problem and the development of effective solution procedures correspond to the two phases of the present research effort. Mathematical models have been developed for the laser gun pose determination and for Ursula's location determination problems. Modifications were made to the existing solution procedures available in the literature (discussed in Chapter 2), to develop effective variants. The solution procedures were evaluated using standard test problems available in literature (discussed in Chapter 4). The models were tested using ideal case scenarios as discussed in Chapter 4. The performance of the solution procedures in real-life situations was studied by generating random, realistic test problems. The results are presented and analyzed in Chapter 5.

Different solution procedures have been investigated to solve Ursula's location determination problem. The Golden Section method, which is based on the alternate formulation discussed in Chapter 3, is the best solution procedure for this problem. In case of changes in model which result in additional independent variables in problem, the BFGS and the Hooke and

Jeeves methods could be the alternate solution procedures. The laser gun pose determination problem is a calibration problem which can be solved by the BFGS and the conjugate gradient (Sherali and Ullular's method using inexact line search) methods. The relative performance of these solution procedures is dependent on the kinematic errors in the setup.

The software modules of the solution procedures were developed using Borland C++ on a PC platform. The modules have a flexible design and can accommodate changes in the problem size and objective function with ease. These software modules will be delivered to the Robotics and Mechanisms group at Virginia Tech. It can be concluded from the present research efforts that an adequate model and effective solution procedures are available for the solution of Ursula's location determination problem.

Future research

Further research beyond the scope of this thesis, may focus on the following possibilities.

1. The hardware setup in the nuclear reactor vessel usually undergoes changes. The proposed model may be extended to accommodate these changes.
2. The error modeling procedure may be modified by adding error to additional parameters.
3. Other types of direction generation and updating schemes for the conjugate gradient methods and the quasi-Newton methods may be investigated.
4. The solution procedures may be integrated into Ursula's calibration and operating software.

5. The formulation and the methodology developed in this thesis may be extended to the robotic calibration and inverse kinematics problems in the field of robotics. The solution procedures could also be applied to other engineering design problems related to the fields of mechanical, aerospace and electrical engineering.

Chapter 7

References

1. Bazaraa, M.S., Sherali, H.D., and Shetty, C.M., **Nonlinear Programming : Theory and Algorithms**, John Wiley & Sons, Inc., New York, NY, 2nd edition, 1993.
2. Craig, J.J., **Introduction to Robotics: Mechanisms and Control**, Addison-Wesley Publishing Company, 1984.
3. Dennis, J.E., Schnabel, R.B., **Numerical Methods for Unconstrained Optimization and Nonlinear Equations**, Prentice-Hall, Inc., New Jersey, 1983.
4. Fallon, J.B., Shooter, S.B., Reinholtz, C.F., and Glass, S.W., "Design of an Underwater Robot for Nuclear Reactor Vessel Inspection", *Proceedings of the ASCE Specialty Conference*, pp 311-319, 1994.
5. Gallerini, R., and Sciomachen, A., "On Using LP to Detect Collision Between a Manipulator Arm and Surrounding Obstacles", *European Journal of Operational Research*, 63, pp 343 - 350, 1993.
6. Gorinevsky, D., "An Algorithm for On-line Parametric Nonlinear Least Square Optimization", *Proceedings of the 33rd Conference on Decision and Control*, pp 2221 - 2226, December 1994.
7. Han, Y.S., Snyder, W.E., and Bilbro, G.L., "Pose Determination using Tree Annealing", *Proceedings of the IEEE Conference on Robotics and Automation*, pp 427 - 432, 1990.

8. Jacoby, S.L.S., Kowalik, J.S., and Pizzo, J.T., **Iterative Methods for Nonlinear Optimization Problems**, Prentice Hall Inc., New Jersey, 1972.

9. Law, A.M., Kelton W.D., **Simulation Modeling and Analysis**, McGraw-Hill, Inc., 1991.

10. Luenberger, D.G., **Introduction to Linear and Nonlinear Programming**, 2nd edition, Addison-Wesley Publishing Company, Amsterdam, 1984.

11. Nash, S., Sofer, A., **Linear and Nonlinear Programming**, McGraw-Hill, Inc., New York, 1996.

12. Nobiki, A., Naruse, H., Yabuta, T., Tateda, M., "Accurate Multi - Viewpoint Stereo Measurement Using Nonlinear Optimization Method", *Proceedings of the 1993 IEEE/RSJ International Conference on Intelligent robots and Systems*, pp 1851 - 1856, July 1993.

13. Pendyala, C.M., "On the Optimal Location of Transmitters for Micro - Cellular Radio Communications System Design", *Masters Thesis*, VPI & SU, 1994.

14. Shanno, D.F., "Conjugate Gradient Methods with Inexact Searches", Mathematics of Operations Research, pp 244 - 256, Vol 3, August 1978.

15. Sherali, H.D., and Ulular, O., "Conjugate Gradient Methods Using Quasi-Newton Updates with Inexact Line Searches", *Journal of Mathematical Analysis and Applications*, pp 359 - 377, 150(2), 1990

16. Stone, H.W., and Sanderson, A.C., "Statistical Performance Evaluation of the S - Model Arm Signature Identification Technique", *Proceedings of the IEEE International Conference on Robotics and Automation*, Vol. 2, pp 939 - 946, 1988.

17. Tidwell, P.H., Hendricks, M.W., Fallon, J.B., Stulce, J.R., Beisgen, L.C.V., Maples, A.B., and Reinholtz, C.F., 1993, “Acoustic Pose Determination Applied to an Underwater Mobile Robot”, *Proceedings of the 3rd National Applied Mechanisms & Robotics Conference*, Cincinnati, Ohio, pp. AMR-93-084-01-06.

18. Voruganti, R.S., “Robot System Characterization: Error Modeling, Identification, Analysis, and Minimization”, *Ph.D. dissertation*, VPI & SU, 1995.

19. Zak, G., Fenton, R.G., and Benhabib, B., “Improvement of Robot Kinematic Parameter Estimation through the Weighted Least Squares Solution Method”, *Proceedings of the 2nd National Applied Mechanisms & Robotics Conference*, pp IIC.2-1-5, 1991.

20. Zak, G., Fenton, R.G., and Benhabib, B., “ A Simulation Technique for the Improvement of Robot Calibration”, *Journal of Mechanical Design*, Vol. 115, No. 3, pp. 674 - 679, 1993.

21. Zhuang, H., Wang, K., and Roth, Z.S., “Optimal Selection of Measurement Configurations for Robot Calibration Using Simulated Annealing”, *Proceedings of the IEEE Conference on Robotics and Automation*, pp 393 - 398, 1994.

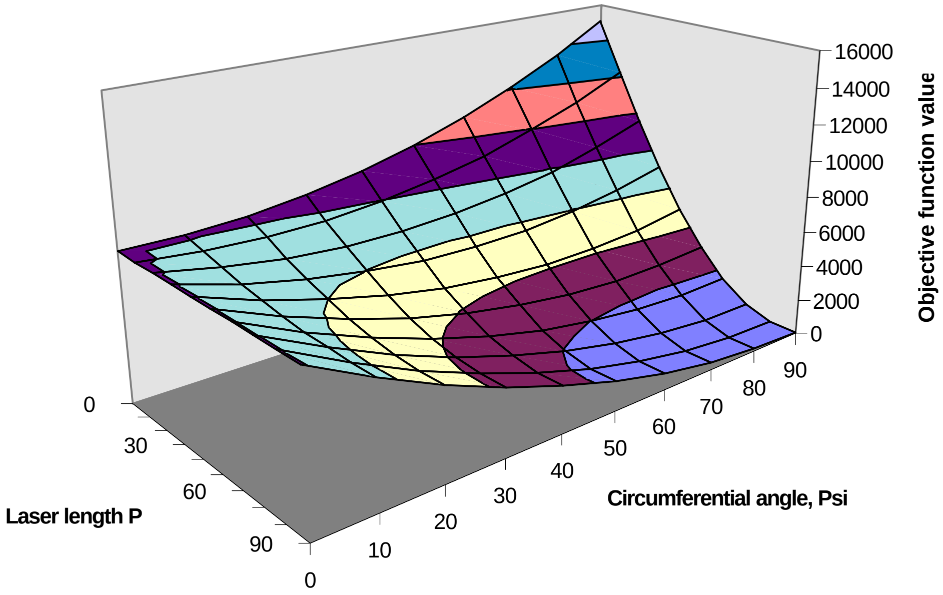
Appendix A

Nominal values and ranges of perturbation for the error parameters

Parameters	**Nominal Values**	**Ranges of perturbation**
Location of the targets:		
Target I:	(80,0,0)	(0.154, 0.154, 0.115)
Target II:	(0,80,0)	(0.154, 0.154, 0.115)
Target III:	(-80,0,0)	(0.154, 0.154, 0.115)
Target IV:	(0,-80,0)	(0.154, 0.154, 0.115)
Laser gun mechanism pose parameters:		
$(x, y, z, \alpha, \beta, \gamma\}$	{0,0,0,0,0,0}	$\{0.25, 0.25, 0.25, 1^o, 1^o, 1^0\}$
Laser gun pose parameters:		
$(x_L, y_L, z_L, \alpha_L, \beta_L, \gamma_L\}$	{0,0,0,0,0,0}	$\{0.02,0.02,0.02, 0.25^o,0.25^o,0.25^o\}$
Pan axis Denavit - Hartenberg notation parameters:		
$(\theta_I, a_0, \alpha_0, d_1\}$	{0,0,0,0}	$\{0.5^o, 0.02, 0.016^o,0\}$
Tilt axis Denavit-Hartenberg notation parameters:		
$(\phi_I, a_1, \alpha_1, d_2\}$	$\{0,0,-\pi/2,0\}$	$\{0.5^o, 0.02, 0.016^o,0\}$
Pan angle and tilt angles		
(θ, ϕ)	-	$\{0.0275^o, 0.0275^o\}$

Appendix B

Objective function contour of a test problem for Ursula's location determination (Problem set I, Seed 5)



Analytically determined minimum of this function has corresponding (P, Psi) = (90.4, 85.69deg)

Vita

Mr. Sai Sivanand Tunuguntla hails from Secunderabad, India. Born on 12th June 1973, he got an undergraduate degree in Mechanical Engineering, from Osmania University (Hyderabad, India) in 1994. He pursued his masters degree in Industrial and Systems Engineering (Operations Research), Virginia Tech, Blacksburg, beginning in fall 94.

Sai S. Tunuguntla