Design of a semi-regular Stewart platform manipulator for a desired workspace

Debabrata Chatterjee[1] and Ashitava Ghosal[2*]

[1] Mechanical Engineering Department, Indian Institute of Science, Bangalore, India

[2] Mechanical Engineering Department, Indian Institute of Science, Bangalore, India

[*] Corresponding author (email: asitava@mecheng.iisc.ernet.in)

Design of a semi-regular Stewart platform manipulator for a desired workspace

Abstract

The SRSPM is the most widely used and hence the most important Gough-Stewart platform configuration for various practical applications. Hence design for geometrical parameters to achieve optimal performance is of considerable interest. This paper presents a search based algorithm to determine the optimal geometrical parameters for the SRSPM for given desired workspace specifications. We have used the knowledge of the generic shape of the workspace as revealed by previous studies, and a few fundamental observations on the effect of varying different geometrical parameters on the geometry of the workspace to design an algorithm to solve our design problem. Two main observations used are -- the size of the end effecter determines what volume of the reachable workspace is actually the region over which the tool is orientable in any desired directions and the actuated length can be increased to have a new increased workspace volume.

Keywords: Gough-Stewart platform, SRSPM, workspace, design, discretization technique

1 Introduction

The Gough-Stewart platform is a very well known and particularly important class of fully parallel manipulators. The generic mechanism consists of two platforms connected by six S-P-U legs, with the prismatic joint (P) being the actuated one and all legs acting in parallel. The mechanism was first proposed by Stewart[1] but the first working prototype of the Stewart's mechanism was made by Gough, hence the name Gough-Stewart platform is used to denote them. Their popularity is due to their superior performance characteristics such as higher stiffness and high precision in positioning. The semi regular Stewart platform manipulator or the SR-SPM was built and studied by Fichter[2]**.** The mechanism derives its name from the special symmetries in its geometry. The SRSPM is also the most widely used configuration finding usage in variety of applications such as flight simulator mechanisms, precise positioning systems, light profile machining etc.

However, as is the case with other parallel manipulators, the analysis of SRSPM is also characterized with complexities such as relatively small workspace with irregular boundary, complex input-output relations and occurrence of singularities in the workspace.

Merlet[3,4] has compiled the various workspace definitions associate with a parallel manipulator. A parallel manipulator motion may be restricted due to different reasons; mechanical limits of active and passive joints, self collision between elements of the robot, singularity regions, which may split the workspace so as to restrict motion across them. The "constant orientation workspace" has been defined as the set of all possible locations of the operating point that can be reached with a given orientation [3]. The "orientation workspace" is all the possible orientations that can be reached while end effecter point is in a fixed location. For parallel manipulators the end effecter is generally desired to achieve only a range of orientations, at all points in its desired region of operation (Merlet[3], Haug[5]). Correspondingly the term "total orientation workspace" is defined, which represents all locations of end effecter point that may be reached with all the orientations among a set defined by ranges on the orientation angles [3]. Hence forth in this paper we have used the term 'workspace' to imply desired 'total orientation workspace'.

Merlet[3, 4] summarizes the analysis approaches for workspace determination. There are three classes -- discretization methods (see, for example, Fitcher[2] and Arai[6]), geometrical methods (see, for example, Gosselin[7]) and Jacobian matrix techniques(see, for example Jo[8]). Fichter[2] used discretisation techniques to identify the workspace of parallel manipulators. In this method an estimated region in space is discreetized to a set of ordered points, and each point is checked for containment within the workspace of the robot. Generally the workspace is determined by incrementally moving the end effecter from one point to adjacent point, then solving the inverse kinematics problem at that point to determine the link lengths. The link lengths are checked for joint limit constraints for verifying the containment of the point in the workspace. The mesh size of the grid used determines the resolution of the workspace boundary evaluated by this method. Geometrical method was first proposed by Gosselin[7], who used this approach to determine the constant orientation workspace of the robot. This approach can be extended to evaluate the total orientation workspace for a defined range of orientation as a subset of the constant orientation workspaces of the defining orientations. Merlet[4] further extended this approach to take all physical constraints into account while determining the workspace. The Jacobian matrix method is different from the other two techniques for evaluation of workspace. While the other two techniques look for joint constraints and other physical constraint violation, in this method the workspace is seen as the region where the velocity Jacobian matrix is not rank deficient. In physical terms it checks for the maneuverability of the end effecter, which is quantified in terms of the volume of the velocity ellipsoid at a point.

The SRSPM architecture for Gough-Stewart platform evolved from the early 3-3 and 3-6 Stewart platform architecture which had coincident connecting points for the legs. This feature severely restricted the joint movements and hence limited the workspace to even smaller volume. The SRSPM with the base platform as a regular hexagon and top platform as a semi regular hexagon came to be regarded as the best option (as the GS platform with both platforms regular hexagons is a special configuration which is not always completely constrained.) Much of the work on Stewart platforms has been focused on analysis and construction of prototypes for various applications in different settings. There are a few works on dimensional synthesis and systematic design. (Dasgupta[9])

The fundamental design problem may be stated as given a workspace volume in space what are the optimal geometric parameters of the robot such that it will contain within its workspace the entire desired volume. This analysis should preferably be done keeping in mind occurrence of singularity barriers as workspaces are often segmented into disconnected regions by them. Gupta[10] discusses different optimization preferences which may arise as per the application the robot is being designed for. In his study Merlet[11] approached the design problem in two steps -- first he identified the feasible domain in the parameter space that will satisfy the workspace requirement, then he conducted a numerical search for optimality of his desired variables in that domain. Lou[12, 13] used controlled random search technique for multi-parameter optimization in design of parallel manipulators, and they used the technique successfully for kinematic design of the SRSPM

The term optimality in our design context, considering only spatial variables and parameters, translates to determining values for geometric variables of the Stewart platform such that they are minimized, without hampering the basic design requirement. The order of precedence of these variables for optimization according to their importance is a design decision.

In this paper we present a numerical search based algorithm to find the geometrical parameters of a SRSPM such that our desired workspace region just fits into its workspace. First, we discretise the given desired workspace region into layers perpendicular to the Z direction. In the second step, we search each layer, from bottom upwards, in a spiral pattern and check if each point in a layer is inside the workspace. If the check fails, then we use certain heuristics, based on work done by earlier researchers, to change the design variables. The algorithm stops when the entire workspace is scanned and it then outputs the optimal values of the design variables.

This paper is organized as follows: in section 2 we describe the geometry and we present a brief review of the inverse kinematics of the SRPSM. In section 3, we present issues related to the design of an SRSPM: first we state in explicit terms what the design problem is; next we present the observations that are used (based on previous studies) on which we have relied to design our algorithm. In addition, in section 3 we have presented the overall method for design, we have described a numerical algorithm to estimate optimal length dimensions of the SRSPM, subsequently we have suggested a strategy to optimize the semi included angles. Finally in section 4 we have provided some numerical results obtained.

2 The Semi Regular Stewart Platform Manipulator.

The construction and kinematics of SRSPM has been discussed amply by Fischer [2]. Here we make a brief review to introduce the notations and concepts used in our design process.

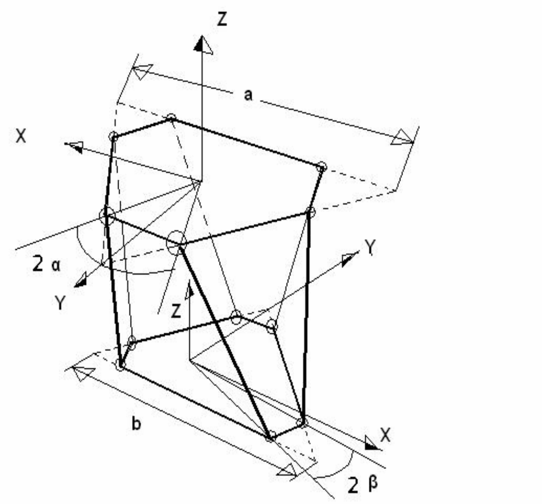

Fig 1: Geometry of a Semi Regular Stewart Platform Manipulator.

The general Stewart platform consists of two platforms connected by six legs which can be actuated to vary their length. Of the two one of them is called the base and other the platform, tool or end effecter. Each leg has one of its end points attached to the base and the other to the platform through spherical joints. A right handed coordinate frame is attached at a convenient location to each of the platforms. Topologically the system is symmetric, and the assignment of names is purely arbitrary. The SRSPM as the name suggest has some special regularities in its geometry, hence the number of independent parameters required to describe the robot adequately comes down to six; As shown in the figure 1 both the base and the platform are conceived out of equilateral triangles of sides **b** and **a** respectively. Another geometric parameter associated with the platforms is the semi included angle at which the ends of the triangles are truncated (symmetrically) to form the hexagons, these are **α** and **β** for the top platform and base respectively. The legs are attached to the platform at the corner of the hexagons in a regular order. These along with the maximum and minimum leg lengths describe the geometry of the SRSPM completely. We have described the leg lengths in terms of two parameters, $\mathbf{L_{min}}$ and $\mathbf{L_{act}}$

which are the minimum leg length and the actuated length of the leg, respectively. The maximum leg length is the sum of $\mathbf{L_{min}}$ and $\mathbf{L_{act}}$.

Inverse Kinematics: Each of the six points of attachment of the legs on the base is described by position vector $\mathbf{M_b(i)}$ with respect to the base coordinate frame. Similarly each of the points on the platform are described by vectors $\mathbf{M_a(i)}$ with respect to the platform frame. The orientation of the platform with respect to the base is given by the rotation matrix **[R]**. The rotation matrix **[R]** can be used to transform the vectors in the platform coordinate frame to the base coordinate frame. The position of the platform with respect to the base is given by the position vector **P** which is the position of the origin of the platform coordinate frame with respect to the base.

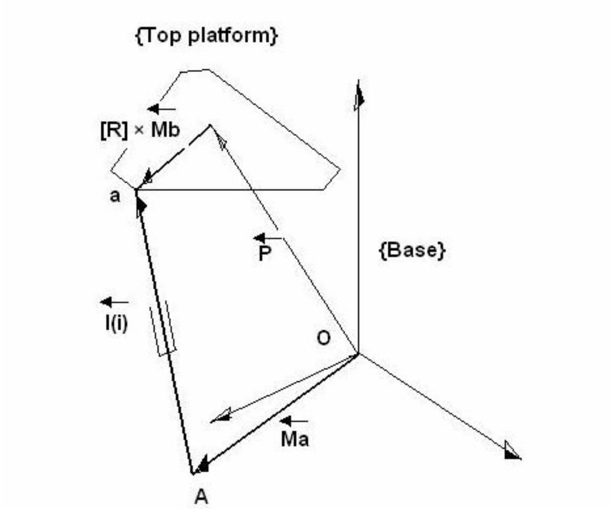

Fig 2: Vector equation to solve the IK problem.

The vector equation relating the four vectors shown in figure 2 is given by:

$$\mathbf{M_a(i) + l(i) = P + [R] \times M_b(i),}\ i=1,\dots,6. \quad (1)$$

or $$\mathbf{l(i) = P + [R] \times M_b(i) - M_a(i)}$$

The length of the vector **l(i)** is the leg length and is obtained as:

$$l(i) = \sqrt{(\mathbf{l(i)} \cdot \mathbf{l(i)})},\ i=1,\dots,6 \quad (2)$$

3 Design of SR SPM

In this section, we describe the design problem for an SRSPM and present some of the key observations based on earlier studies. These observations are used in our algorithms.

3.1 Problem definition

The design problem can be stated as given a well defined volume in space and range of desired orientations the end effecter is expected to achieve, find optimal values for geometric parameters of the SRSPM so that the workspace of the designed GS platform contains within itself the entire volume of the defined volume of desired workspace. In precise terms, we are given (1) the range of orientation (in Z-Y-X Euler angles) defining the orientation workspace of interest, and (2) geometry of the desired workspace volume as a rectangular box with base dimension **d**×**d** and height **h.** The sides of the base square are aligned along the X and Y axis of the base coordinate frame, and the intersection of the diagonals of the base square is at (0, 0, **H**) with respect to the base coordinate frame. The problem is to obtain optimal values for (1) platform triangle side lengths **a** and **b**, (2) included angles **α** and **β,** and (3) maximum actuated leg length $\mathbf{L_{max}}$ and minimum actuated leg length $\mathbf{L_{min}}$. These last two terms me be expressed in terms of the variables $\mathbf{L_{min}}$ and $\mathbf{L_{act}}$, as:

$$\mathbf{L_{min}=L_{min}}, \qquad \mathbf{L_{act} = L_{max} - L_{min}}.$$

3.2 Observations

Observation 1 (On the nature of workspace): Gosselin[7] made the following observation on the nature of the constant orientation workspace: The constant orientation workspace in three dimension Cartesian space is obtained by the intersection of regions bounded by spheres, and the boundary of the (constant orientation) workspace will consist exclusively in a set of portions of spheres.

The argument can be extended to make the following observation: as the workspace is the intersection set of all the constituent constant orientation workspaces, hence its boundary must be contained in selected sections of spheres bounding the individual constant orientation workspaces.

Du Plessius [14] used optimization techniques for high resolution mapping of the workspace boundary. The shape of the boundary as revealed by their study supports the above observations. The workspace is a region in space about the Z axis of the coordinate frame bounded by sections of spheres. In our algorithm we have fitted our desired rectangular region in this workspace region.

Observation 2: Kumar[15] discussed the dependency of orientibility of the end effecter on the size of the end effecter. In the case of a Gough-Stewart platform, theoretically, when the end effecter tends to be infinitesimally small, it is orientable in any desired direction with infinitesimal actuator movements. In that case the entire reachable workspace of the robot is actually the workspace over which it is totally orientable. However keeping base dimension fixed to some finite value if we now increase the size of the top platform to some finite value, the actuator movement required to achieve any and all orientations at some given representative point is proportional to the top platform size. In this new configuration the workspace is different from the reachable workspace and now is only a portion of it. The end effecter is no longer orientable at all points to where it can reach due to limitation on lengths of actuator movement.

From the above observation it can be said that keeping **a** constant if **b** is increased, the workspace volume increases as the relative size **a/b** decreases and the end effecter becomes more orientable.

3.2 Algorithm for estimating optimal values of 'b' and 'actuator length'.

From this algorithm we get an estimate of the optimal values for **b** and $\mathbf{L_{act}}$ required for satisfying the desired workspace specifications. The algorithm is shown as a flow chart in figure 3 and is discussed in detail in this section.

It may be noted that without any loss of generality, we can normalize all length dimensions i.e. **b**, $\mathbf{L_{act}}$, $\mathbf{L_{min}}$ as well as length dimensions of required workspace i.e. **d**, **h** and **H**, with respect to **a**; alternately we can set **a** as1.

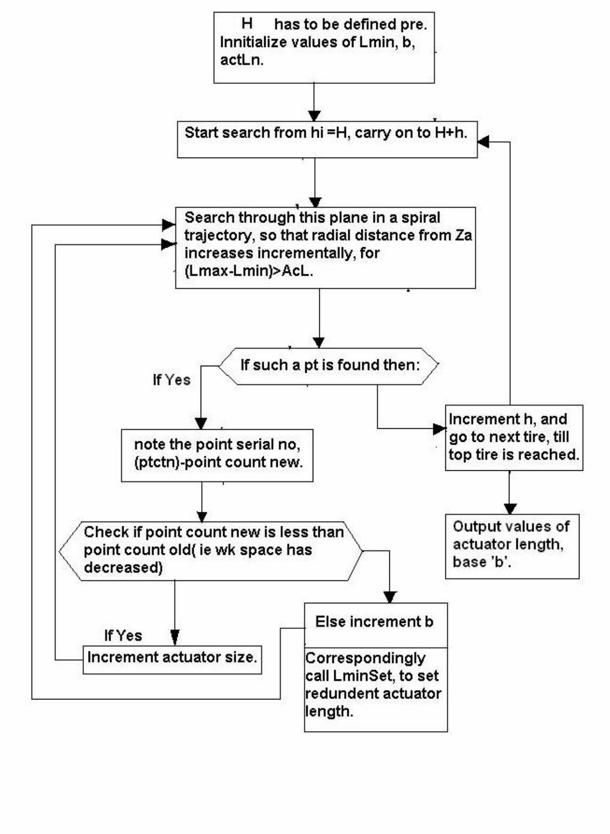

Fig 3: Flowchart describing the algorithm.

Step 1: Set values of **a**, **b** to 1, $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are set to some intermediate convenient values if not already specified. **Step 2:** Initialize values of $\mathbf{L_{min}}$ and $\mathbf{L_{act}}$ such that it is assured the workspace of the SRSPM with **a**=**b**=1 configuration contains entirely within itself the bottom plane of the rectangular desired workspace region. No optimization takes place at this stage and it is only an initialization subroutine so that it is assured at least some point of our required volume is contained in the workspace of our initial configuration robot. This is important because our search must start at a point which is contained inside the workspace and that when we move on to the adjacent point and check for containment, our previous point acts as a reference for comparison, and when we have a point where the containment check fails, we know we have crossed a workspace boundary.

However there can be other reasonable initializations. We need to ensure that our start point for search must be assured to be "well" inside the workspace volume of the initial configuration SPM (so that when an increment in **b** or $\mathbf{L_{act}}$ is made it does not come out from the workspace). The workspace is discretised vertically along the Z direction into layers. At each layer the algorithm follows a spiral trajectory to move from the centre outwards point by point checking for containment of the point within the workspace region of the SRSPM. **Step 3:** Move to the next layer and search along the spiral trajectory from centre outwards for containment of the path points.

To check containment of a point, we solve the inverse kinematics problem at that point, for all required range of orientations. (In effect we have taken combination of the extreme values of the Euler angles.) The inverse kinematics problem is solved to get the leg lengths. The maximum leg length is checked for achieve-ability, because achievable maximum actuator length is limited. If for the then set actuator length (and redundant actuator length) the maximum required leg length is achievable then it implies the GS platform can be assembled for all required configurations at that point and hence this point lies within the workspace.

At this layer if it is ensured all points are contained within the workspace only then we move on to the next layer and carry on the same search as described in this step.

If we come across a point where containment criteria is violated we go to step 4.

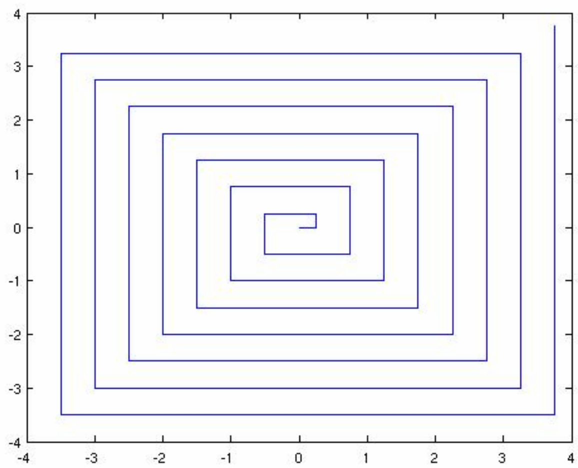

Fig 4: A typical spiral search path.

Step 4: This is the incrementing step. As we have reached a point which is not contained within our present workspace volume, we increase the workspace to fit this point into it. We have two options namely: we can increase **b** so that workspace volume increases as explained in observation 2, or we increase $\mathbf{L_{act}}$ so that the total reach of the end effecter increases, hence the workspace also increases. We choose the options according to the logic described below. At a layer if it is the first time we are coming across a point which violates the containment criteria, we increment **b**. Incrementing **b** has two effects, first in certain directions workspace dimensions increase another effect is that actuator lengths remaining constant, the reachable workspace goes down towards Z=0 plane.

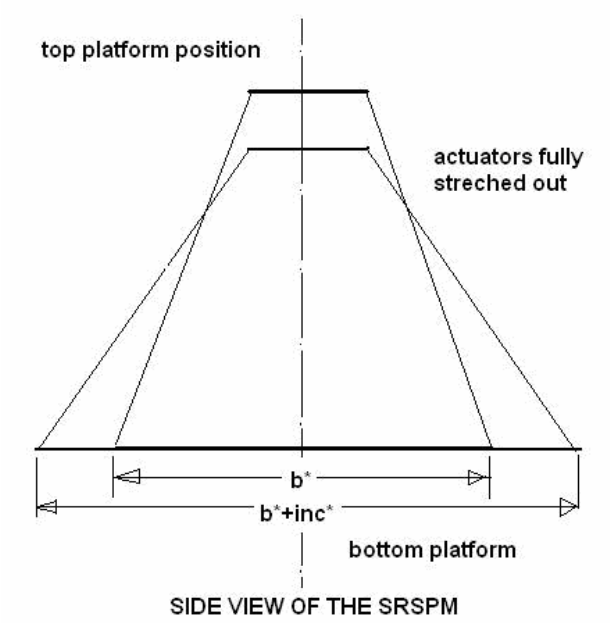

Fig 5: Illustration of workspace going down towards Z=0 plane.

At the same time we note the radial distance from Z axis (or as we have done some index proportional to it) at which the increment was made. Then we restart our search at that layer but from x=y=0.

At the next instance we come across a point (at that layer) which violates containment criteria, we check the radial distance at which it occurs and compare the radial distance with the previous value. If the radial distance has decreased, then we can say no amount of increment in **b** will do, in that case we increase our total workspace by incrementing (active) actuator length. Or else we increment **b**. After any increment we start from the center, at that layer, and repeat the search.

Every time we change **b**, we make a check to see with existing joint limits if the bottom plane continues to be wholly within the workspace. If we find any portion of the bottom plane has moved out of our present workspace we make adjustments in $\mathbf{L_{min}}$ (decrease suitably to a new value) and then carry on with the search where we had left it.

This search proceeds through the entire region of interest, from the bottom layer to the top layer, till the entire desired region is included in the workspace.

The values of **b** and $\mathbf{L_{act}}$ to which the algorithm converges are the required dimensions; their optimality is justified from the fact that they are values at which the desired rectangular region just fits into the workspace of the robot. However the quality of fit is dependent on the choice of incrementing values for **b** and $\mathbf{L_{act}}$ as well as choice for distance of spacing between adjacent points used while discretization of the workspace.

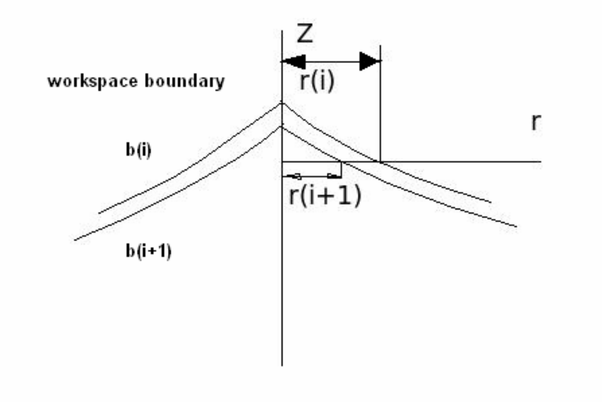

Fig 6: The case of workspace boundary being encountered at a lesser radial distance, an increment in **b** been made in the previous step.

Justification of the incrementing scheme: Figure 5 is an illustration showing the GS platforms with actuators fully stretched, with different base dimensions **b**. Incrementing **b** causes the reachable workspace to go down. In figure 6 we have shown a representative section of the workspace in the vertical plane for two different values of **b**, all other parameters remaining constant. The peak of the workspace goes down with increment in **b**. **r(i)** is the radial distance from Z axis at which workspace boundary was encountered at the i^{th} search at that level, **r(i+1)** is the corresponding length at $i+1^{th}$ search, after an increment in **b** was made at the end of i^{th} search. The figure illustrates the case when no amount of incrementing in **b** will suffice to expand the workspace to include the desired point at **r(i).** It may be noted that the workspace may not necessarily move down with every increment in **b**, we have only described a scheme which can identify the condition when incrementing **b** alone will not suffice.

3.3 Optimizing values of α and β.

From the previous algorithm we have an initial estimate of **b** and $\mathbf{L_{act}}$. To analyze the effect of **α** and **β** on the workspace size and hence to obtain optimal values for **α** and **β** we note that theoretically though the semi included angles may vary from 0° to 60°, convenient values can be chosen from space availability point of view. We have chosen representative values of 10° and 45° as the lower and upper limit for these variables. Now we define a subroutine that can evaluate the workspace boundary at any given Z value, for any required values of **α** and **β**. This is in fact the intersection line of the 3D workspace boundary with he z=**c** plane (located at height **c** from base origin).

To evaluate the boundary we shoot a ray at an angle θ with respect to (+)Y axis. We move along this ray from r=0, in increments, checking if the GS platform can be assembled at this point i.e. we check for containment of this point in the workspace. When it comes across a point along this line where it can no more be assembled, we say the previous point was on the workspace boundary. Varying θ from 0° to 360°, we can get a mapping of the workspace boundary at that height.

The topmost layer is the critical layer where the square section just fits into the workspace boundary. We have tried to find out the combination of **α** and **β** for which we have the largest workspace section at this layer. We can map the workspace boundaries for different combinations of **α** and **β**. The combination of **α** and **β** for which we get the largest workspace area at this layer is our optimal values for these variables.

We have taken half diagonal length of the best fit square with centre at (x=0, y=0) as our objective function for maximization. This can be obtained by working out the radial distance of intersection of rays shooting out from the origin along the diagonal directions and the workspace boundary, and taking the minimum of the four distances we get corresponding to the four rays.

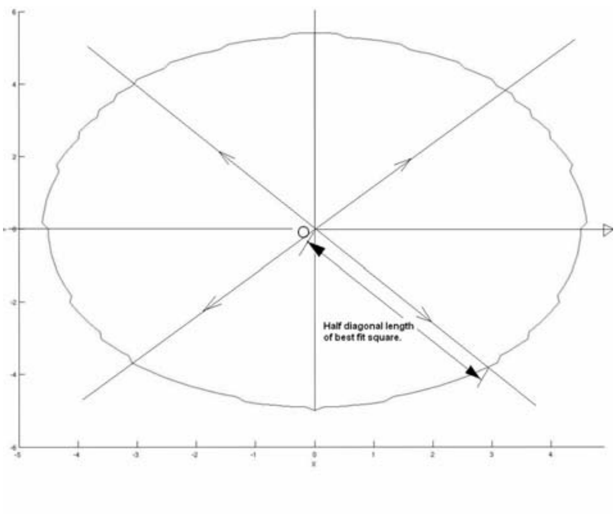

Fig 7: Half diagonal length for the best fit square.

We discretise the feasible domain of **α** and **β** and over this domain we numerically compute the value of the objective function. We take the values of **α** and **β** for which the objective function is maximized. Hence we have the largest possible workspace for the given **b** and $\mathbf{L_{act}}$.

Finally with the new optimized values of **α** and **β,** we re-implement our previous search algorithm to arrive at better estimates for values of **b** and $\mathbf{L_{act}}$.

4 Results

The numerical schemes outlined in this paper were developed and implemented using Matlab® 7.7 in Linux environment. As has been mentioned earlier on all length dimensions are normalized with respect to **a** (**a**=1), i.e., for real dimensions we can set **a** to our required value in any convenient units, and multiply all other length dimensions with appropriate multiplying factor. The base coordinate frame is attached at the centroid of the base platform triangle, similarly the top platform coordinate frame is attached at the centroid of the top platform triangle; the coordinate frames are so aligned that their respective Z axis are perpendicular to the respective platform planes. Their Y axises pass through one of the vertices of their respective platform triangles. At home position the platforms are parallel to each other, but the coordinate frames are so oriented with respect to each other that a 180° rotation about the Z axis of the base frame matches its orientation with that of the platform frame.

For discretising the workspace in the X-Y plane interval length chosen, for generating points of the spiral, was 0.25. Along the vertical direction segmentation was made in intervals of 0.2. Values of **b** were incremented by 0.2 at a time; values of $\mathbf{L_{act}}$ were incremented by 0.2 at a time.

For **α** and **β** optimization the domain of **α** between 10 °to 45° and **β** between 10° and 45° were discretized in intervals of 5° for both the variables.

Problem 1: Design a SRSPM given:

1. Orientation range of top platform desired: In Z-Y-X Euler angles; Z may vary from -22.5° to +22.5°, Y may vary from -22.5° to +22.5°, X may vary from -22.5° to +22.5°.
2. Desired workspace dimensions: Base square side d=6.6, vertical height h=1.8 located at height H=2.5.

Solution: for first estimation values of **α** and **β** are set to 12°, 15° respectively. Initializing values for $\mathbf{L_{min}}$ = 2.3606, $\mathbf{L_{act}}$ = 3.4013.

Estimates of optimal values after first implementation of algorithm:

a = 1, **b** = 4.20, $\mathbf{L_{min}}$ = 2.36 and Lact =5.00.

Estimates of optimal values of **α** and **β** after implementing domain search for largest workspace:

α = 10° and **β** = 10°.

Final estimates for length dimensions with new values for **α** and **β** are

a = 1, **b** = 4.20, $\mathbf{L_{min}}$ = 2.36, and $\mathbf{L_{act}}$ = 5.00.

Problem 2: Design a SRSPM given:

1. Orientation range of top platform desired: In Z-Y-X Euler angles; Z may vary from -22.5° to +22.5°, Y may vary from -22.5° to +22.5°, X may vary from -22.5° to +22.5°.
2. Desired workspace dimensions: Base square side d=5, vertical height h=3 located at height H=2.5.

Solution: For first estimation values of **α** and **β** are set to 30°, 36° respectively. Initializing values for

$\mathbf{L_{min}}$ = 4.9394 and $\mathbf{L_{act}}$ = 2.3633.

Estimates of optimal values after first implementation of algorithm:

a = 1, **b** = 5.40, $\mathbf{L_{min}}$ = 2.36, and $\mathbf{L_{act}}$ =5.17.

Estimates of optimal values of **α** and **β** after implementing domain search for largest workspace:

α = 10° and **β** = 10°.

Final estimates for length dimensions with new values for **α** and **β** are

a = 1, **b** = 6.00, $\mathbf{L_{min}}$ = 2.36 and $\mathbf{L_{act}}$ = 5.47.

Note: - The minimum radial distance evaluated was 3.400 and the maximum 3.5750, and the variation is quite small. The workspace boundary evaluated by the method is highly irregular with number of marked points of sharp unevenness. These factors have probably contributed to the anomalies in the final results. The anomaly may be removed by using finer discretisation.

5 Conclusion

In this paper we presented a systematic method for spatial design of a SRSPM. The algorithm presented in this paper can easily be extended to cases of different workspace geometry. Also using improved methods for workspace mapping, and using 'regularized workspaces' for containment evaluation, performance and reliability of the algorithm and the overall method can be improved.

Reference

[1] D. Stewart, *"A platform with six degrees of freedom."* Proceedings of the Institution of Mechanical Engineers, 1965, pp. 371–386.

[2] E. F. Fichter, *"A Stewart platform based manipulator: general theory and practical construction."* The International Journal of Robotics Research, Vol. 5, 1986, pp. 157–182.

[3] J. -P. Merlet, *Parallel Robots,* Springer, 2001.

[4] J.-P. Merlet, "*Determination of the orientation workspace of parallel manipulators.*" Journal of Intelligent and Robotic Systems, Vol. 13, 1995, pp. 143–160.

[5] E.J. Haug, J.-Y. Wang, J.K. Wu. "*Dexterous workspaces of manipulators. I. Analytical criteria.*" Mechanics of Structures and Machines, Vol.20, 1992, pp.321–361.

[6] T. Arai, K. Cleary, T. Nakamura., H. Adachi., K. Homma, "*Design, analysis and construction of a prototype parallel link manipulator.*" IEEE International Workshop on Intelligent Robots and Systems, 1990, pp. 205-212.

[7] C. M. Gosselin, "*Determination of workspace of a 6 degree of freedom parallel manipulator.*" ASME Journal of Design, Vol. 112, 1990, pp. 331-336.

[8] D.-Y Jo, E.J Haug, "*Workspace analysis of multibody mechanical systems using continuation methods.*" Transactions of the ASME Journal of Mechanisms, Transmissions, and Automation in Design, Vol.111, 1989, pp. 581–589.

[9] B. Dasgupta, T. S. Mruthyunjaya , "*The Stewart platform manipulator: a review.*" Mechanism and Machine Theory, Vol.35, 2000, pp. 15-40.

[10] K. C. Gupta, B. Roth, "*Design considerations for manipulator workspace.*" Transactions of ASME, Journal of Mechanical Design, Vol. 104(4), 1982, pp. 704–712.

[11] J.-P. Merlet, *"Workspace-oriented methodology for designing a parallel manipulator.*" In Proceedings of IEEE International Conference on Robotics and Automation, 1996, pp. 3726 - 3731.

[12] Y.Lou, J. Liu, N.Chen, Z.Li., *"Optimal design of parallel manipulators for maximum effective regular workspace"*, IEEE/RSJ International Conference on Intelligent Robots and Systems, 2005, pp 795- 800.

[13] Y. Lou, G. Liu, J. Xu, Z. Li, *"A general approach for optimal kinematic design of parallel manipulators,"* In Proceedings of IEEE International Conference on Robotics and Automation, 2004, pp. 3659-3664.

[14] L. J. Du Plessis and J. A. Synman, *"A numerical method for the determination of dexterous workspace of a GS platform."* International Journal for Numerical Methods in Engineering, Vol. 52, 2004, pp. 345-369.

[15] A. Kumar and K.J. Waldron, "*The workspace of a mechanical manipulator.*" Transactions of the ASME Journal of Design, Vol.103, 1981, pp. 665-672.