A FULLY GEOMETRIC APPROACH FOR THE WORKSPACE AREA OF THE GANTRY-TAU PARALLEL KINEMATIC MANIPULATOR

Ilya Tyapin
The University of Queensland
Brisbane, Australia QLD 4072
Email: ilya@itee.uq.edu.au

Geir Hovland
Agder University College
N-4898 Grimstad, Norway
Email: geir.hovland@hia.no

Torgny Brogårdh
ABB Robotics
SE-721 68 Västerås, Sweden
Email: torgny.brogardh@se.abb.com

ABSTRACT
One of the main advantages of the Gantry-Tau machine is a large accessible workspace/footprint ratio compared to many other parallel machines. The design presented in this paper achieves a workspace/footprint ratio of more than 2.7 with fixed length links, which is high for a PKM. Typical PKMs, for example the Delta robot, has a ratio less than one. Optimisation schemes are typically used in order to achieve a maximum ratio. In this paper we present a new fully geometric approach for calculating the workspace area of the Gantry-Tau. A fully geometric approach is faster and more accurate than numeric approaches and of significant benefit in kinematic optimisation schemes.

KEY WORDS
workspace optimisation, geometric approach, parallel kinematic manipulator.

1 Introduction

The Tau family of parallel kinematic manipulators (PKM) was invented by ABB Robotics, see [1]. The Gantry-Tau was designed to overcome the workspace limitations while retaining many advantages of PKMs such as low moving mass, high stiffness and no link twisting or bending moments. For a given Cartesian position of the robot each arm has two solutions for the inverse kinematics, referred to as the left- and right-handed configurations. While operating the Gantry-Tau in both left- and right-handed configurations, the workspace will be significantly larger in comparison with both a serial Gantry-type robot and other PKMs with the same footprint. The intended application of the robot is for machining operations requiring a workspace of a typical serial-type robot, but with higher stiffness.

In this paper we consider the triangular-link variant of the 3-degree-of-freedom (DOF) Gantry-Tau structure, which was first presented in [2]. Triangular mounted links give several advantages: they enable a reconfiguration of the robot, a larger reach is obtained in the extremes of the workspace. When using parallel links, the orientation of the manipulated platform will be constant, which increases the risk of collisions of the arms with the manipulated platform in the extremes of the workspace area.

In [3] an optimisation method for the 3-DOF Gantry-Tau with no triangular links was presented. Two geometrical parameters of the machine were optimised to maximise the cross-sectional workspace area. Our paper is an extension of the work in [3]. The new contributions of this paper are: the optimisation is made for the Gantry-Tau with triangular mounted links, the optimisation is made over the whole workspace volume and not just the cross-sectional workspace area and finally a geometric approach to find the cross-sectional workspace area of the Gantry-Tau is presented and introduced into the design optimisation. Also, the joint angle limits are taken into account in our paper.

In the past many researchers focused on the calculation of the workspace area of PKMs. In order to calculate the workspace one can employ discretisation methods, geometrical methods or analytical methods. For the discretisation method a grid of nodes with position and orientation is defined. Then the kinematics is calculated for each node and it is straightforward to verify whether the kinematics can be solved and to check if joint limits are reached or link interference occurs. The discretisation algorithm is simple to implement but has some serious drawbacks. It is expensive in computational time and results are limited to the nodes of the grid. One example of this approach is [4].

Analytical workspace area methods investigate the properties of the kinematic transmission [5, 6]. Most approaches are based on the inverse kinematics because it can typically be solved in closed form and it is easer to distinguish between multiple solutions. Whereas discretisation methods are based on the full inverse kinematics, the analytic approaches usually only require parts of the inverse kinematics. Even so, analytical methods can be expensive in computational time.

Using geometrical methods the workspace can be calculated as an intersection of simple geometrical objects [7], for example spheres. The midpoints of the spheres are found from the pivot points for example PUS (prismatic joint, universal joint, spherical joint) and UPS (legs form a cylinder and a sphere). Constraints on the joints and on the articulated coordinates can be introduced through further restrictions on the geometrical entities and the intersection of these objects can be done with techniques that are known from computer aided design (CAD). Examples of such constraints are floors, ceilings and prismatic joint locations to avoid collisions with the links.

The works presented in [8] and [9] are examples of partial geometric approaches to calculate the workspace of

a 3-DOF PKM. They are only partial geometric because numeric integration is used to calculate the workspace. Design optimisation was attempted by both [8] and [9]. In addition, [9] presents the relationships between the workspace and link lengths of all planar 3-DOF parallel manipulator.

Another interesting work is the paper by [10]. In that paper a fully geometric approach to calculate the reachable workspace was presented for a 6-DOF Hexapod type PKM. The differences compared to our paper is the use of variable link lengths instead of fixed actuators at the robot base. Also, [10] use an inverse kinematics (rotational matrix) to define a workspace. Similar work is also presented in [11].

Most researchers use a basic inverse kinematics (rotational matrix) or Jacobian matrix as a part of a geometric approach to define a workspace. In our paper the workspace is defined as a set of simple geometrical shapes (segments of a circle, triangles, polygons). All calculations are based on cross points between basic 2D geometrical figures (line, circle, polygon).

In this paper a geometrical method for the Gantry-Tau that uses parameterisation of the midpoints of the three spheres that intersect at the TCP is used. The geometric approach developed in this paper also handles the fact that the platform orientation is not constant, which is also a difference from most of the existing published work. Section 2 briefly describes the kinematics and the configuration of the triangular-link version of the Gantry-Tau. The geometrical workspace calculations of the reachable workspace are described in Section 3. The influence of the joint angle limits on the reachable workspace is presented in Section 4. Finally, the optimisation problem is formulated in section 5 and some results are presented in Section 6.

2 Kinematic Description

The 3-DOF Gantry-Tau is a reconfigurable PKM and is illustrated in Fig. 1. The figure shows the PKM structure in both the left-handed and right-handed configuration (also called assembly modes). The Tau structure is characterised by a clustering of the links in groups of 1, 2 and 3, respectively, with fixed link lengths L_1, L_2 and L_3. Three linear actuators are used at the base to move the three arms independently in the global X direction. More details about the structure and kinematics of the Gantry-Tau can be found in [2] and [12]. The actuator track locations are fixed in the Y and Z directions and the locations are denoted T_{1y}, T_{1z}, T_{2y}, T_{2z}, T_{3y} and T_{3z}, respectively. The dimensioning of the PKM's support frame in the case it is symmetric is given by the two variables Q_1 and Q_2 as illustrated in Fig. 1, where Q_1 is the depth and Q_2 is the height. The width of the machine in the X direction is given by the length of the actuators. The goal of this paper is to simultaneously optimise the work space volume with respect to the design parameters Q_1, Q_2, L_1, L_2 and L_3, where L_i are the arm lengths according to Fig. 1. Fig. 2 shows the fixed kinematic parameters of the moving platform, which

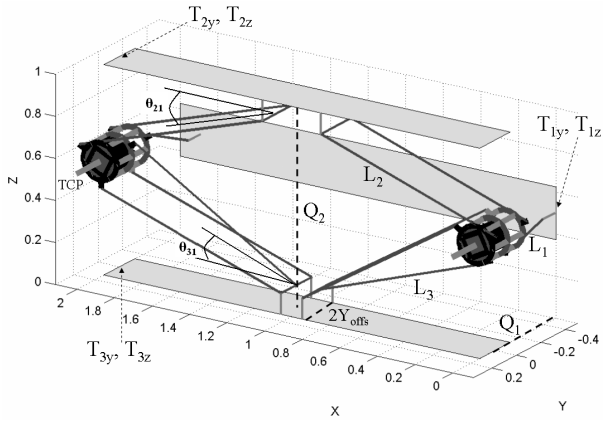

Figure 1. *The 3-DOF reconfigurable Gantry-Tau robot.*

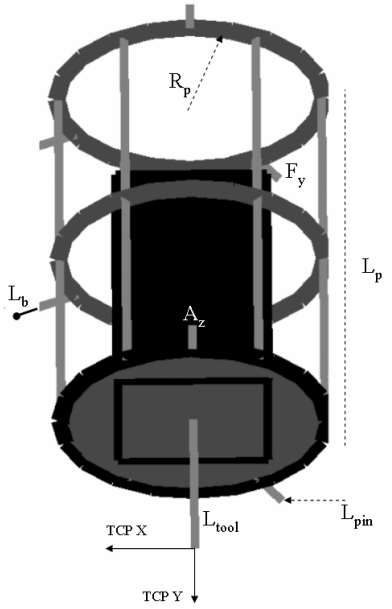

Figure 2. *The manipulated platform of the Gantry-Tau robot.*

are not included in the design optimisation. L_p is the platform length, R_p is the platform radius, L_{tool} is the tool length, L_{pin} is the length from the platform circle of radius R_p to the connection point for the universal joints and L_b is the length from the connection point to the centre of the joint. A_z and F_y are the Z and Y coordinates in the TCP coordinate frame of two connection pins on the platform. The single link with length L_1 is connected to the pin F via a universal joint pin, while one of the links with length L_2 is connected to the point A. These two coordinates are required in the following section.

3 Workspace Area and Volume Functions

The workspace in the X-direction is limited by the length of the linear actuators. Care must also be taken to avoid collisions between the arms, platforms and actuators. Fig. 3 shows three circles, one for each arm. The centres of the

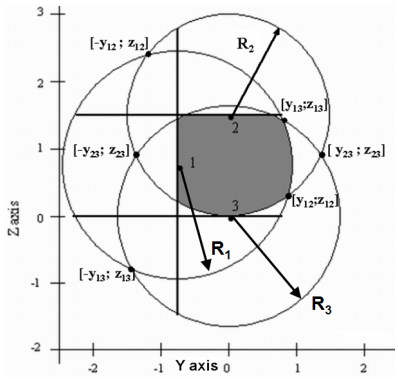

Figure 3. *The cross-sectional workspace area of the Gantry-Tau in the YZ-plane.*

circles are located at the actuator positions. The radii of the circles equal the arm lengths plus the distance from the end of the arm to the TCP. For example, the radius of circle 2 equals L_2+A_z. The TCP can only reach points inside all of the circles. Fig. 3 also contains three solid lines in the YZ-plane. The TCP is not allowed to move outside these lines because they indicate the positions of the support framework. Note that it is possible to get outside these lines at line parts where the links and platform do not collide with the linear actuators, for example at $Y > 0.2$ and at a small platform also with Y about -0.3. However, these areas are omitted in the analysis in this paper.

The valid TCP positions are illustrated in grey in Fig. 3. In the following text, an analytic closed-form solution is presented for calculating the grey area of Fig. 3. The arm lengths plus the distances from the ends of the arms to the TCP are denoted $L_1^{''}$,$L_2^{''}$, $L_3^{''}$ and equal to the radii of the circles. The lower and upper actuator limits are X_L and X_H. The cross-sectional area in Fig. 3 is valid when $x \in [X_L; X_H]$, where X_L and X_H are defined in Section 6.

The fixed y- and z-coordinates of the track positions T_{iy} and T_{iz} where $i = 1, 2, 3$ are given below. Note that the optimisation parameters Q_1, Q_2, L_1, L_2 and L_3 are introduced here, demonstrating where the track positions T_{2z}, T_{1y} and T_{1z} depend on Q_1 and Q_2. The track positions are defined in Fig. 1.

$$T_{3y} = 0 \quad T_{3z} = Z_{offs} \tag{1}$$
$$T_{2y} = T_{3y} \quad T_{2z} = T_{3z} + Q_2 - Z_{offs} \tag{2}$$
$$T_{1y} = T_{3y} - Q_1 + Z_{offs} \tag{3}$$
$$T_{1z} = \frac{(T_{2z} + T_{3z})}{2} \tag{4}$$

The midpoints of the spheres in the YZ-plane are then given by: sphere 1: $[T_{1y}, T_{1z}]$, sphere 2: $[T_{2y}, T_{2z}]$ and sphere 3: $[T_{3y}, T_{3z}]$. Given the midpoints, the functions that describe the circles are given below.

$$(T_{1y} - y)^2 + (T_{1z} - z)^2 = L_1^{'2} \tag{5}$$
$$(T_{2y} - y)^2 + (T_{2z} - z)^2 = L_2^{'2} \tag{6}$$
$$(T_{3y} - y)^2 + (T_{3z} - z)^2 = L_{3d}^{'2} \tag{7}$$

L_{3d} which is slightly shorter than L_3 is the distance from actuator 3 to the mid-point of the triangular-mounted link pair at the platform side. Note the notation with the single superscripts $'$ which will be introduced later in this section.

The y- and z-coordinates of the cross-points between two circles are defined in four stages. The procedure presented below is more flexible and faster than an algorithm based on the cross points calculation directly from the Fig. 3. Some advantages of the presented algorithm are: small number of operations (additions, subtractions, multipications) and flexibility if the circle parameters are changed (radii, coordinates of the centres). The stages are specified below and illustrated in Fig. 4.

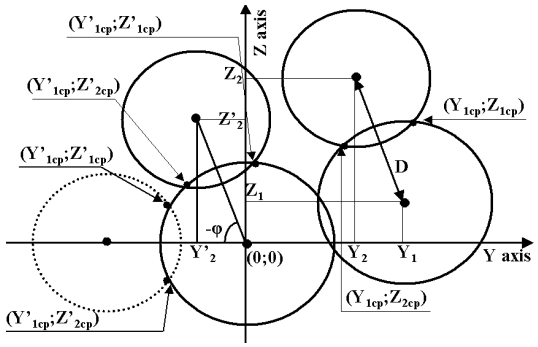

Figure 4. *Illustration of the definition of the cross points between two circles.*

Stage 1: The centre of the first circle is shifted to the point $(0; 0)$. The centre of the second circle is shifted the same distances on the Y- and Z axes. The calculation of the new centres of the circles are given below.

$$Y'_1 = Y_1 - Y_1 \quad Z'_1 = Z_1 - Z_1 \tag{8}$$
$$Y'_2 = Y_2 - Y_1 \quad Z'_2 = Z_2 - Z_1 \tag{9}$$

Where Y_1 and Z_1 are the original coordinates of the first circle centre, Y_2 and Z_2 are the original coordinates of the second circle centre, Y'_1, Z'_1, Y'_2, Z'_2 are the new coordinates of the circles.

Stage 2: The second circle is rotated by an angle φ until the z-coordinate of the second circle centre equals 0, see Fig. 4. The equations for the cross points between the circles are then given below.

$$D = \sqrt{(Y_2 - Y_1)^2 + (Z_2 - Z_1)^2} \tag{10}$$
$$\cos\varphi = \frac{Y'_2}{D} \quad \sin\varphi = \frac{Z'_2}{D} \tag{11}$$
$$Y'_{1cp} = \frac{R_1^2 + D^2 - R_2^2}{2D} \quad Z'_{1cp} = \sqrt{R_1^2 - Y'^2_{1cp}} \tag{12}$$
$$Z'_{2cp} = -\sqrt{R_1^2 - Y'^2_{1cp}} \tag{13}$$

where D is the distance between the circle centres, φ is the angle of rotation, R_1 and R_2 are radii of the first and second circles, $(Y'_{1cp}; Z'_{1cp})$ and $(Y'_{1cp}; Z'_{2cp})$ are coordinates of the cross points between the first and the second circle after rotation.

Stage 3: The cross points between the circles after rotation were found in step 2. In this step the cross points between the circles before rotation are defined.

$$Z''_{1cp} = Y'_{1cp} \sin\varphi + Z'_{1cp} \cos\varphi \quad (14)$$
$$Y''_{1cp} = Y'_{1cp} \cos\varphi - Z'_{1cp} \sin\varphi \quad (15)$$
$$Z''_{2cp} = Y'_{1cp} \sin\varphi + Z'_{2cp} \cos\varphi \quad (16)$$
$$Y''_{2cp} = Y'_{1cp} \cos\varphi - Z'_{2cp} \sin\varphi \quad (17)$$

$(Y''_{1cp}; Z''_{1cp})$ and $(Y''_{2cp}; Z''_{2cp})$ are the cross points between the circles before rotation.

Stage 4: The original cross points between the circles are found from stage 1 to stage 3.

$$Z_{1cp} = Z''_{1cp} + Z_1 \quad Y_{1cp} = Y''_{1cp} + Y_1 \quad (18)$$
$$Z_{2cp} = Z''_{2cp} + Z_1 \quad Y_{2cp} = Y''_{2cp} + Y_1 \quad (19)$$

Where $(Y_{1cp}; Z_{1cp})$ and $(Y_{2cp}; Z_{2cp})$ are the original cross points between the circles.

The y-coordinates of the cross points between circle number 2 and 3 are illustrated in Fig. 3 as y_{23} and $-y_{23}$. If y_{23} is not a real number, then there is no common cross-point between circles 2 and 3. In this case, the cross-sectional area will be zero. The early detection of a zero workspace is an advantage of a geometric workspace area approach compared to a solution based on the inverse kinematics. If y_{23} is real, then the cross-points between circles $1, 3$ and $1, 2$ and $2, 3$ can be defined from eqs. (8-19).

The y-coordinates of the cross-points between circles 1 and 3 are y_{13} and $-y_{13}$. The y-coordinates of the cross-points between circles 1 and 2 are y_{12} and $-y_{12}$.

The workspace is divided into 4 areas A_1, A_2, A_3 and A_4 as illustrated in Figs. 5, 6, 7 and 8. Areas A_2 and A_3 are divided into 3 small areas (two of them are shown in grey colour and the third is situated between the grey areas). Area A_1 has 2 small areas, which are triangular (grey area) and a segment of the circle. Area A_4 is a quadrilateral (grey area). Area A_3 in Fig. 7 consists of two segments of

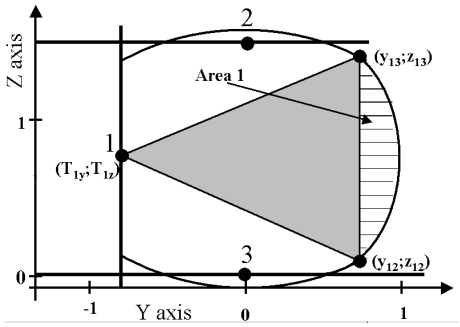

Figure 5. *Illustration of workspace area 1.*

the circle 3 for small values of Q_2 because the upper limit of the workspace is partly circular and partly a straight line. At bigger values of Q_2 the upper limit will be the circle 3 arc between the points $(Q_1; z_{3Q_1})$ and $(y_{13}; z_{13})$. The equations for the area A_3 are given below in a few steps.

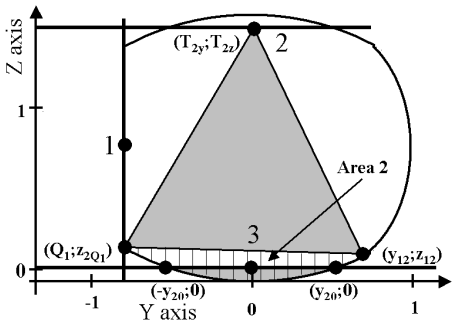

Figure 6. *Illustration of workspace area 2.*

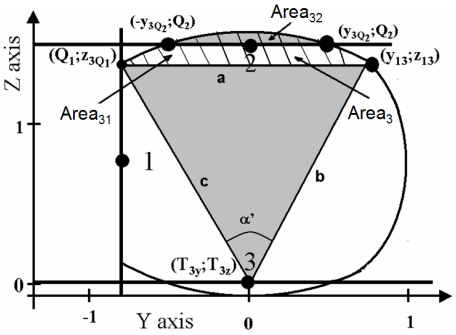

Figure 7. *Illustration of workspace area 3.*

Step 1: Additional cross points are necessary. The cross points between a circle and a line $z = Const$ are defined as

$$y_1 = y + \sqrt{R^2 - (z_1 - z)^2} \quad (20)$$
$$y_2 = y - \sqrt{R^2 - (z_2 - z)^2} \quad (21)$$
$$z_1 = Const \quad z_2 = Const \quad (22)$$

Where z and y are coordinates of a circle centre, y_1 and y_2 are y-coordinates of the cross points between a circle and a line, R is a radius of a circle. For the circle 3: $z = T_{3z}$, $y = T_{3y}$, $y_1 = y_{3Q_2}$, $y_2 = -y_{3Q_2}$ and $R = L''_3$. The points are shown in Fig. 7.

The cross points between a circle and a line $y = Const$ are defined as

$$z_1 = z + \sqrt{R^2 - (y_1 - y)^2} \quad (23)$$
$$z_2 = z - \sqrt{R^2 - (y_2 - y)^2} \quad (24)$$
$$y_1 = Const \quad y_2 = Const \quad (25)$$

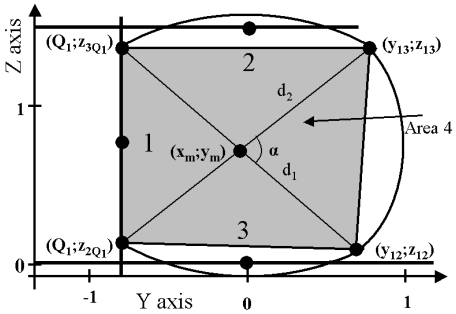

Figure 8. *Illustration of workspace area 4.*

The z and y are coordinates of a circle centre, z_1 and z_2 are z-coordinates of the cross points between a circle and a line. The line R is a radii of a circle. For the circle 3 : $z = T_{3z}$, $y = T_{3y}$, $z_1 = z_{3Q_1}$ and $R = L_3^{''}$. All these points are shown in Fig. 7.

The cross points $(-y_{3Q_2}; Q_2)$ and $(y_{3Q_2}; Q_2)$ are found from eqs.(20-22). In addition, the cross point $(Q_1; z_{3Q_1})$ is found from eqs.(23-25), point $(y_{13}; z_{13})$ is found from eqs.(8-19) as a cross point between two circles.

Step 2: The area of a circle segment is given below.

$$a = \sqrt{(y'_2 - y'_3)^2 + (z'_2 - z'_3)^2} \quad (26)$$

$$b = \sqrt{(y'_1 - y'_3)^2 + (z'_1 - z'_3)^2} \quad (27)$$

$$c = \sqrt{(y'_2 - y'_1)^2 + (z'_2 - z'_1)^2} \quad (28)$$

$$\alpha = \arccos\left(\frac{b^2 + c^2 - a^2}{2bc}\right) \quad (29)$$

$$AreaTR = \frac{1}{2}bc\sin(\alpha) \quad (30)$$

$$Area_{segment} = r^2 * \frac{\alpha}{2} - AreaTR \quad (31)$$

Where point $(y'_1; z'_1)$ is a centre of a circle, points $(y'_2; z'_2)$ and $(y'_3; z'_3)$ are the limits for the arc of the circle (chord), a, b and c are the side lengths of the triangle abc which is shown in Fig. 7 as a grey area. α is an angle between two lines which are coming from the centre of a circle to the points on the arc. The angle α is always positive. $AreaTR$ is an area of a triangle abc. $Area_{segment}$ is an area of the segment of a circle limited by two points on the circle arc. For the circle 3 : $y'_1 = T_{3y}$, $z'_1 = T_{3z}$, $y'_2 = Q_1$, $z'_2 = z_{3Q_1}$, $y'_3 = y13$, $z'_3 = z13$. $Area_{segment} = Area_{31}$. Lengths a, b, c, angle α and the area of a triangle are clearly shown in Fig. 7.

Step 3: This step depends on the optimisation parameter Q_2. If Q_2 is smaller than $L_{3d}^{''}$, a second segment of the circle 3 will be found and $Area_{segment} = Area_{32}$. This segment is situated on the top of the solid line $z = Q_2$. In Fig. 7 the second segment is a grey area. The second segment of the circle 3 will be found from eqs.(26-31), but the chord limits on the arc are $(-y_{3Q_2}; Q_2)$ and $(y_{3Q_2}; Q_2)$. If Q_2 is greater than $L_3^{''}$, the second segment equals 0.

Step 4: The area A_3 is defined as an area of the segment of the circle 3 : $A_3 = Area_{31} - Area_{32}$.

The areas A_1 and A_2 will be found in the same way as area A_3. There is no second segment for the area A_1.

The area A_4 is the area of a quadrilateral. The coordinates of the vertexes will be found from eqs.(8-22). The area of the quadrilateral is given below.

$$\begin{aligned} A_4 &= \frac{1}{2}(x''_1 y''_2 - x''_2 y''_1) + (x''_{n-1} y''_n - x''_n y''_{n-1}) \\ &+ (x''_n y''_1 - x''_1 y''_n) \end{aligned} \quad (32)$$

where coordinates $(x''_n; y''_n)$ are coordinates of a vertex of a polygon and n is the number of polygon vertexes.

The total cross-sectional area A_{Total} is given by

$$A_{Total} = A_1 + A_2 + A_3 + A_4 \quad (33)$$

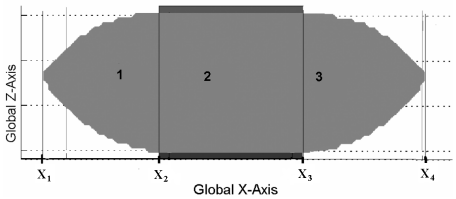

Figure 9. *The workspace area of Gantry-Tau machine in the XZ-plane when it is reconfigured to work in both positive and negative x-direction.*

The next step is to calculate the workspace volume as a function of the cross-sectional areas. The geometric shape of the workspace volume is difficult to describe by geometrical shapes. Hence, the volume function is calculated partially numerically as described below.

$$V = \sum_{x=x_1}^{x=x_4} A_{Total}(x)\delta \quad (34)$$

where δ is a step of the summation and $A_{Total}(x)$ is a workspace area for the given x-coordinate.

First, the upper and lower bounds x_1 and x_4 for TCP on the X-axis must be found, and following this, additional bounds x_2 and x_3, that depend on the kinematics must be determined. These additional bounds are the actuator limits $x_2 = X_L$ and $x_3 = X_U$, illustrated in Fig. 9. Following this, new radii for the three circles have to be found, one for each robot arm. The new radii depend on the given x position. Finally, the cross-sectional workspace area can be calculated. Some different workspace areas for the given x positions are presented in Fig. 10

The 4 bounds x_1, x_2, x_3, x_4 define 3 sections on the global X-axis. The sections are $[x_1; x_2], [x_2; x_3], [x_3; x_4]$. These 4 sections are numbered in Fig. 9. The TCP can only reach points with x-coordinates between x_1 and x_4. Between these limits, the cross-sectional area varies as illustrated in Fig. 9 and Fig. 10.

The radii for the three circles in Fig. 10 depend on the x-coordinate. For the section $x \in [x_1; x_2]$ the radii are defined as follows.

$$L'_1 = \sqrt{L''_1 - (x - x_2)^2} \quad L'_2 = \sqrt{L''_2 - (x - x_3)^2} \quad (35)$$

$$L'_{3d} = \sqrt{L''_3 - (x - x_3)^2} \quad (36)$$

Obviously, the radii will shrink for the given section. For example at the position $x = -1.1$ the area A_4 will become a triangular area instead of the quadrilateral area for $x = 1$. The single superscript $'$ as mentioned earlier denotes the section-dependent circle radii. The double superscript $''$ denotes the nominal circle radii. The workspace volumes for the sections $x \in [x_1; x_2]$ and $x \in [x_3; x_4]$ are the same and do not have to be computed numerically twice.

In the second section $x \in [x_2; x_3]$ the radii equal the nominal values, ie.

$$L'_1 = L''_1 \quad L'_2 = L''_2 \quad L'_{3d} = L''_3 \quad (37)$$

The methods for calculating the circle intersection points and the areas $A_1 - A_4$ apply to all cross-sectional areas between x_1 and x_4. Fig. 10 shows different workspace areas for 6 different x-coordinates and for the section $[x_2; x_3]$, in which the cross-sectional area is constant.

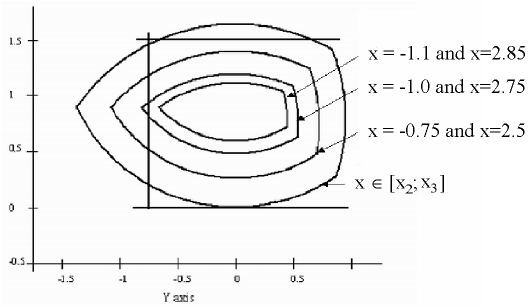

Figure 10. *The YZ-plane workspace as a function of x.*

4 Joint Angles

In this section the influence of the physical limits of the joint angle on the workspace of the Gantry-Tau is analysed. The design of the Gantry-Tau includes revolute joints for the links. For example, link 1 has a rotation around the Z-axis with a joint limit $\theta_{11} = \pm 180$ degrees and around the X-axis with a joint limit $\theta_{12} = \pm 33$. The links in arms 2 and 3 have rotations around the Y-axis with a joint limit $\theta_{21} = \theta_{31} = \pm 180$ degrees and around the X-axis with a joint limit $\theta_{22} = \theta_{32} = \pm 33$. The influence of the joint angle limit on the workspace will be found separately for each arm. In Fig. 11 the influence of the joint limits for arm 3 around the X axis is shown. In Fig. 12 the influence of the joint limits for arm 1 around the X axis is presented. The grey areas are lost areas of the maximum workspace if the joint angle limits are less than required limits for the maximum workspace. For example, the limits must be greater or equal to $\theta_{22} = \theta_{32} = \pm 40$ degrees for the Gantry-Tau machine presented in this paper. If the joint angles limits are smaller, the workspace area will be reduced compared to the maximum possible workspace. The distance on

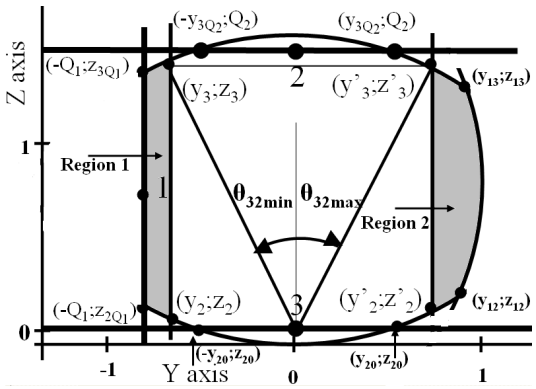

Figure 11. *Illustration of the influence of the joint angle limits on the workspace area for arm* 3 *around the X axis*

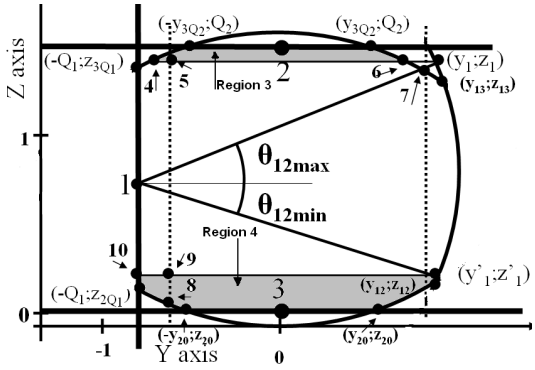

Figure 12. *Illustration of the influence of the joint angle limits on the workspace area for arm* 1 *around the X axis*

the Y-axis $(y_3 - T_{3y})$ will be defined from the right triangle $(T_{3y}; T_{3z})(T_{3y}; z_3)(y_3; z_3)$. The distance on the Z-axis $(z_3 - T_{3z})$ will be found from the same triangle. The formulas for the distances are given below.

$$Hyp_3 = L''_{3d}$$
$$y_3 = Hyp_3 * \sin(\theta_{32}) \quad z_3 = Hyp_3 * \cos(\theta_{32})$$

where Hyp_3 is the hypotenuse of the right triangle and θ_{32} is the joint angle limit.

The calculation of the distances will give us the coordinates of the cross point between the circle 3 and the hypotenuse of the right triangle. The coordinates $(y'_3; z'_3)$ are defined in the same way.

The reachable area including the influence of the joint angles limits is shown in Fig. 11 and Fig. 12. The grey areas are the parts of the maximum workspace unreachable for the TCP because of the joint angle limits. Up to four extra regions (unreachable for the TCP) have to be found.

The reachable workspace are found from the equation $A = A_{max} - A_{unreachable}$. The unreachable workspace $(A_{unreachable})$ is equal to the set of up to four regions described below. The maximum possible workspace is described in section 3. The algorithm for the unreachable regions is given below.

The first region will be found as a summation of two circle segments and one polygon. The first segment end-points are $(-Q_1; z_{3Q_1})$, $(y_3; z_3)$ and the radius equals the radius of circle 3. The second segment end-points are $(-Q_1; z_{2Q_1})$, $(y_2; z_2)$ and the radius is the radius of circle 2. The polygon vertexes are the end points of the segments. The first region is shown in Fig. 11. If the y-coordinate of the point $(y_3; z_3)$ is greater than the y-coordinate of the point $(-y_{3Q_2}; Q_2)$, the polygon has the extra vertex at point $(-y_{3Q_2}; Q_2)$ and the segment of the circle 3 has the end-point at $(-y_{3Q_2}; Q_2)$. The same modification is applicable for the points $(-y_{20}; z_{20})$ and $(y_2; z_2)$. The segment and polygon calculation algorithm was described in section 3.

In the general case the second region will be defined as a summation of three circle segments and one polygon. The segments end-points are $(y'_3; z'_3)$ and $(y_{13}; z_{13})$ for the

circle 3, $(y'_2; z'_2)$ and $(y_{12}; z_{12})$ for the circle 2, $(y_{12}; z_{12})$ and $(y_{13}; z_{13})$ for the circle 1, The polygon vertexes are the end-points of the segments. The second region will be changed if y'_2 is greater than y_{12} and y'_3 is greater than y'_{13}. In that case the second region consists of the first circle segment only. One or two extra polygon vertexes will be found if the y-coordinates of the points $(y'_3; z'_3)$ and $(y'_2; z'_2)$ are less than the y-coordinates of the cross points $(y_{3Q_2}; Q_2)$ and $(y_{20}; z_{20})$.

The third and fourth regions are shown in Fig. 12. These regions consist of circle segments and polygons. The end points of the circle segments and polygon vertexes are different and depend on the joint angle limits for the arms 1, 2, 3. The algorithm is very difficult to be described in this paper because of limited space. All important control points are shown in Fig. 12. The formulas for the circle segments and polygons were presented in section 3.

In Fig. 1 the angle limits θ_{31} and θ_{21} are shown. These angles have one upper limit instead of two (maximum and minimum). For example, if the given joint angle limits for the angles θ_{31}, θ_{21}, θ_{11} are ± 40, the algorithm will change the angles to $0 - 80$.

If the limits described in the previous paragraph are less than 90 degrees, they could take an influence on the reachable workspace of the Gantry-Tau and the arms could not reach the upper limits. The influence on the workspace may be found in a few steps.

Step 1: According to the joint angle limits θ_{31}, θ_{21}, θ_{11} the new radii for the reachable workspace circles (see Fig. 3) will be found. The equations for the new radii are given below.

$$L''_1 = L_1 sin(\theta_{11}) + TCP_1$$
$$L''_2 = L_2 sin(\theta_{21}) + TCP_2$$
$$L''_3 = L_{3d} sin(\theta_{31}) + TCP_3$$

where L''_1, L''_2, L''_3 are the new radii for the circles, L_1, L_2, L_3 are the arm lengths without the TCP, TCP_1, TCP_2, TCP_3 are the distances from the arm ends to the TCP.

Step 2: The new radii will be added to the algorithm for the reachable workspace calculation.

Step 3: The reachable workspace area will be found. The difference between the two workspaces is shown in Fig. 13. The radii $L''_{3original}$ is the original radii of the circle 3 without the joint angle limit taken into account or the joint angle limit is larger or equal to $0 - 90$. The radii $L''_{3changed}$ is the radii of circle 3 if the joint angle limit is taken into account and less than $0 - 90$. The grey area shows the difference between the workspaces if the radius for arm 3 is less than the maximum possible.

5 Optimisation Problem

The workspace volume optimisation is based on the least-squares nonlinear optimisation function *lsqnonlin* in Matlab. The optimisation problem is expressed as

$$\max V(Q_1, Q_2) \tag{38}$$

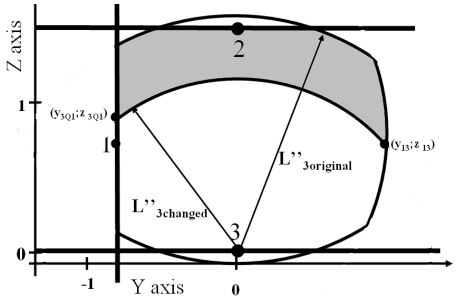

Figure 13. *Illustration of the influence of the joint angle limits on the workspace area for the arm* 3.

$$\text{subject to } Q_1 > 0 \quad \text{and } Q_2 > 0$$

V is a nonlinear function containing squares and square-roots of Q_1 and Q_2. Hence, the total workspace volume as a function of the two support frame design parameters Q_1 and Q_2 will be maximised. Alternatively, a cross-sectional workspace area in the YZ plane could be maximised as in [3]. The volume is preferred as the cross-sectional area depends on the X position. For the optimisation problem described above, the arm lengths are fixed constants. One or more of these lengths can be made unconstrained and part of the optimisation problem. However, if all the arm lengths are unconstrained and have no upper limits, the workspace volume will go to infinity. In practice, long arm lengths have to be avoided to achieve the required machine stiffness.

6 Results

The kinematic parameters of the platform and the support frame parameters were chosen as follows.

$$L_{tool} = 0.003 \quad R_p = 0.088 \quad L_{pin} = 0.028 \quad L_b = 0.03$$
$$L_p = 0.25 \quad X_L = 0 \quad X_H = 2.0 \quad Y_{\text{offs}} = 0.125$$

The final optimisation design parameters of the machine are as follows.

$$Q_1 = 0.5497m \quad Q_2 = 1.1682m \quad V = 2.9038m^3$$
$$L_1 = L_2 = L_3 = 1.2m$$

These results would have been difficult to obtain by a manual design, as all the link lengths are different and Q_2 is different from $2Q_1$ which has been a typical manual design choice of the Gantry-Tau in the past, see [2] and [12].

The required installation space of the Gantry-Tau equals $(X_H - X_L)Q_1Q_2$ which equals $I = 1.2783m^3$ for the optimised design. Hence, the total workspace to installation space ratio for the optimised design is $\frac{V}{I} = 2.7135$ which is large compared to most other PKMs which typically have a ratio of less than one.

7 Conclusion

In this paper a new geometric approach for describing the cross-sectional workspace area of the triangular-link version of the Gantry-Tau has been developed. The functional dependency of the cross-sectional workspace area on the robot's x-coordinate is used to calculate the total workspace volume. This workspace volume is maximised by solving a nonlinear least squares optimisation problem. The optimisation yields two design parameters which determine the placement of two of the linear actuators with respect to the third actuator.

The geometric approach is typically faster than an approach based on a discretisation of the workspace and using the inverse kinematics to check reachability in each discretised workspace cell. In addition, the geometric method returns the exact workspace areas, whereas the accuracy of the discretisation approaches depends on the level of discretisation. If the workspace is discretised more densely to improve accuracy, the computational benefits of the geometric approach increases further.

The Table 1 shows the computational requirements for the three different approaches on the triangular version of the 3-DOF Gantry-Tau PKM described in [2], [12] and [13]. A first approach to calculating workspace is to use the Jacobian matrix derived from the inverse kinematics and matrix inversion. A second approach was described in [13] and based on a half geometric and numerical method. A third approach was presented in this paper. The computing time has been normalised to 1 for the third approach. Because of the significant computational benefits of the geometric approach, this method is far better suited for inclusion in a design optimisation framework compared to discretisation or analytical workspace approaches. The optimised design presented in this paper was obtained in less than an hour on a standard desktop computer using Matlab's Optimization Toolbox. Future extensions of the work presented in this paper will introduce performance criteria such as Cartesian stiffness and resonance frequencies into the design optimisation.

Method	Time
Inverse kinematics	26
Half numerical	2.9
Geometric	1

(39)

Table 1. *Computation time for three different methods.*

References

[1] T. Brogårdh. "PKM Research - Important Issues,as seen from a Product Development Perspective at ABB Robotics", *Proc. Workshop on Fundamental Issues and Future Research Directions for Parallel Mechanisms and Manipulators*, Oct. 3-4, 2002, Quebec, Canada.

[2] T. Brogårdh, S. Hanssen and G. Hovland, "Application-Oriented Development of Parallel Kinematic Manipulators with Large Workspace", *Proc. of the 2nd Intl. Coll. of the Collaborative Research Center 562: Robotic Systems for Handling and Assembly*, Braunschweig, May 2005, pp. 153-170.

[3] S. Johannesson, V. Berbyuk and T. Brogårdh, "A New Three Degrees of Freedom Parallel Manipulator", *In Proc. of the 4th Chemnitz Parallel Kinematics Seminar*, Germany, April 20-21 2004, pp. 731-734.

[4] A.K. Dashy, S.H. Yeoy, G. Yangz and I.H. Chery, "Workspace analysis and singularity representation of three-legged parallel manipulators", in *Proc. of the 7th Intl Conf in Control, Autom., Rob. And Vision*, Singapore, 2002, pp. 962-967.

[5] J. Angeles, "On the Numerical Solution of the Inverse Kinematics Problem", *Int. J. Robotics Res.*, Vol. 4, no. 2, pp.21-37, 1985.

[6] A. A. Goldenberg, B. Benhahib and R. G. Fenton, "A Complete Generalized Solution to the Inverse Kinematics of Robots", *ITEE Journal Robotics Automat*, Vol. RA-1, no. 1, pp.14-20, 1985.

[7] J.-P. Merlet, "Designing a Parallel Manipulator for a Specific Workspace", *Institut National de Recherche en Informatique et en Automatique*, Tech. Rep 2527, 1995.

[8] D. Chablat and P. Wenger, "Architecture Optimisation of a 3-DOF parallel mechanism for machining applications, the Orthoglide", *ITEE Transactions on Robotics and Automation*, Vol. 19, No. 3, pp. 403-410, 2003.

[9] X.-J. Liu, J.-S. Wang, F. Gao, "On the Optimum Design of Planar 3-DOF Parallel Manipulators with Respect to the Workspace", *Proc. of the Intl. Conf. on Robotics and Automation*, San Francisco, April, 2000.

[10] D. I. Kim, W. K. Chung, Y. Youm, "Geometrical Approach for the Workspace of 6-DOF Parallel Manipulators", *Proc. of the Intl. Conf. on Robotics and Automation*, Albuquerque, New Mexico, April, 1997.

[11] I. A. Bonev, J. Ryu, "A Geometrical Method for Computing the Constant-Orientation Workspace of 6-PRRS Parallel Manipulator", *Mechanism and Machine Theory*, Vol. 36, No. 1, pp. 1, 2001.

[12] I. Williams, G. Hovland and T. Brogårdh, "Kinematic Error Calibration of the Gantry-Tau Parallel Manipulator", *Proc. of the IEEE Intl. Conf. on Rob. and Autom.*, Orlando, May 2006.

[13] I. Tyapin, G. Hovland and T. Brogårdh, "Kinematic Optimisation of the Gantry-Tau Parallel Kinematic Manipulator with Respect to its Workspac", *Proc. of the 2006 Austr. Conf. on Rob. and Autom.*, Auckland, NZ.