Computers in Industry 23 (1993) 59–64
Elsevier





# The intersection method: A new approach to the inverse kinematics problem

G.P.A. Alards and Th.E. Schouten

*Department of Informatics / Real Time Systems, Katholieke Universiteit Nijmegen, Nijmegen, The Netherlands*

Currently, there are three different approaches to the inverse kinematics problem, each of which has some disadvantages. The algebraic methodology suffers from the fact that the mathematical trees hide the wood of the solution, making it hard for the user to select the proper solution from the theoretically possible ones. On the other hand, the iterative approach is computationally more demanding and cannot guarantee convergence under all circumstances. Finally, the geometric methodology can only handle relatively simple manipulators.

The problem of calculating the inverse kinematics can be restated as: "given a target position of the end effector of a manipulator in a Cartesian coordinate system, calculate for all joints all possible positions". From these positions all possible sets of joint angles which place the end effector in its target position can be easily calculated.

The new "intersection" method is based on intersection of joint-axis ranges $R(i,j)$: all possible positions of joint-axis$_j$ relative to any other joint-axis$_i$. This joint-axis$_i$ can be the base, the end effector or any joint axis between them. One can regard $R(i,j)$ as the set of possible positions of a reduced manipulator which has only $(j-i-1)$ joints. Note that one works both in the forward and in the backward direction of the original manipulator.The method is intuitively clear, easy to understand and is capable of finding all feasible solutions. It is applicable to fairly complex manipulators such as the PUMA, as will be shown.

*Keywords:* Robotics; Inverse kinematics

## 1. Introduction

Currently, there are three different approaches to the inverse kinematics problem. Each of them has some disadvantages. The algebraic methodology [1] suffers from the fact that the mathematical trees hide the wood of the solution, making it hard for the user to select the proper solution from the theoretically possible ones. On the other hand, the iterative approach [2] is computationally more demanding and cannot guarantee convergence under all circumstances. Finally, the geometric methodology [3] can only handle relatively simple manipulators.

Section 2 of this paper presents a new method [4] of calculating the inverse kinematics that is based on intersection of joint-axis ranges. By nature, it should be called geometrical, but for the fact that this term is already in use. The method is intuitively clear and easy to understand and is capable of finding all feasible solutions. It is applicable to fairly complex manipulators such as the PUMA, as will be shown in Section 3. The final section contains some conclusions and preliminary observations on efficiency and speed.

## 2. Discussion of the intersection method

### 2.1. Description

The problem of calculating the inverse kinematics for a manipulator with $n$ joints can be restated as: "given a target position of the end effector of a manipulator in a Cartesian coordinate system, calculate for all joints all possible positions". From these positions all possible sets of joint angles which place the end effector in its target position can be easily calculated. With

*Correspondence to:* Dr. Theo E. Schouten, Department of Informatics/Real Time Systems, Katholieke Universiteit Nijmegen, P.O. Box 9010, 6500 GL Nijmegen, The Netherlands. E-mail ths@cs.kun.nl

position, both a 3-dimensional point and orientation are meant.

For each joint-axis$_j$ one can define ranges $R(i,j)$: all possible positions of joint-axis$_j$ relative to any other joint-axis$_i$. This joint-axis$_i$ can be the base, the end effector or any joint axis between them. One can regard $R(i,j)$ as the set of possible positions of a reduced manipulator which has only $(j-i-1)$ joints. Its base is joint-axis$_i$ of the original manipulator and its end effector is joint-axis$_j$ of the original manipulator. Note that one works both in the forward and in the backward direction of the original manipulator.

The set $P_{b,j,e}$ of possible positions of joint-axis$_j$ is equal to the intersection of $R(b,j)$ and $R(e,j)$: $P_{b,j,e} = R(b,j) \cap R(e,j)$, where $b$ is the base and $e$ the end effector. For each chosen position for joint-axis$_j$ out of $P_{b,j,e}$, the inverse kinematics problem is then split into two problems with a smaller complexity. The intersection method can then be applied to the reduced manipulators $R(i,j)$ and $R(j,k)$ where $i$, $j$ and $k$ have fixed positions. This algorithm can be repeated until all necessary joint-axis positions are calculated.

### 2.2. Summary of definitions and notations

*Manipulator related*

$b$ — the index of the base ($b = 0$)

$e$ — the index of the end effector ($e = n + 1$)

$i$ — the index of a fixed position$_i$ ($b \leq i < j$)

$j$ — the index of joint-axis$_j$ ($b < j < e$)

position$_j$ — $((x_j, y_j, z_j), v_j)$, where $(x_j, y_j, z_j)$ are the coordinates of the centre of joint-axis$_j$ and $v_j$ is the direction of joint-axis$_j$.

$R(i,j)$ — all possible positions of a joint-axis$_j$ relative to joint-axis$_i$, defining a reduced manipulator

$P_{i,j,k}$ — the set of possible positions of joint-axis$_j$ between the fixed positions $i$ and $k$.

*Geometrical*

$C_2(p,r)$ — all coordinates on the circle in two-dimensional space with centre $p = (x,y)$ and radius $r$

$CI_2(p,r)$ — $C_2(p,r)$ and all points inside it

$C_3(p,r,n)$ — all coordinates on the circle in three-dimensional space with centre $p = (x,y,z)$, radius $r$ and normal $n = (n_x, n_y, n_z)$

$S(p,r)$ — the sphere in three-dimensional space with centre $p = (x,y,z)$ and radius $r$

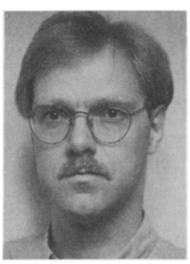
### 2.3. Formulation of the algorithm

The following recursive algorithm (in pseudocode) shows how the intersection method can be used to arrive at a complete solution of the inverse kinematic problem.

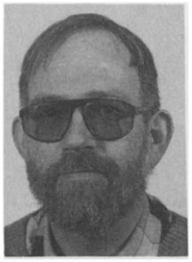
```
begin
  Calculate_Joint_Positions(b,e);
  Calculate_Joint_Angles( );
end

Procedure Calculate_Joint_Positions(i,k)
begin
  Choose j such that i < j < k
  Calculate R(i,j)
  Calculate R(k,j)
  P_{i,j,k} := R(i,j) ∩ R(k,j)
  For each position_j ∈ P_{i,j,k}:
    Calculate_Joint_Positions(i,j);
    Calculate_Joint_Positions(j,k);
end
```

*Remarks*

(1) The choice of $j$ depends on the actual configuration, the simplicity of the resulting calcula-



**Gerie P.A. Alards** received a Master's degree in Computer Science from the Catholic University of Nijmegen, The Netherlands in 1989. Shortly after his graduation he joined AT&T Network Systems Netherlands. His current research interests are in the areas of software development automation and neural networks.



**Theo E. Schouten** received his PhD in High-Energy Physics from the University of Nijmegen, The Netherlands. After positions at NIKHEF (Amsterdam), CERN (Geneva) and the Max Planck Institute (Munich), he is since 1984 a staff member of the Informatics Department of The University of Nijmegen. His current research is in the areas of (satellite and medical) image processing, real-time systems and neural networks.

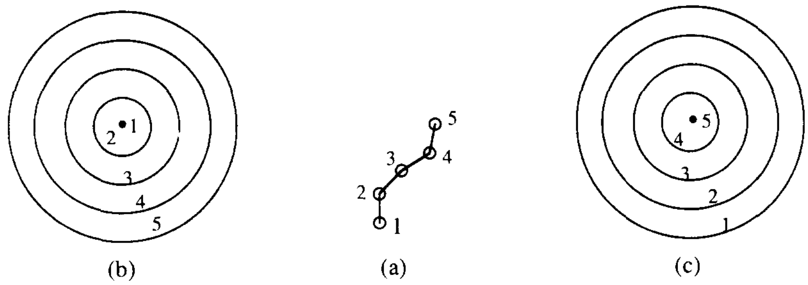



Fig. 1. Simple manipulator with four revolute joints in a plane.

tion or constraints on the links of the manipulator. This point will be clarified by examples.

(2) It is not always necessary to calculate the entire subspace $P_{i,j,k}$. When the objective is just to find one feasible position, this step can be combined with the next one to yield a specific choice of position$_j$.

(3) The choice of position$_j$ is completely free, at least theoretically. In practice, some optimization using e.g. the current position$_j$ of joint-axis$_j$ may suggest that a certain position is to be preferred.

## 3. Examples

### *3.1. A manipulator with four revolute joints in a plane*

In this simple case the manipulator (Fig. 1(a)) moves in a plane and all the directions of the joint-axes in each configuration are parallel to each other and each link has length $L$. Figure 1(b) shows the joint-axis ranges relative to joint-axis$_1$ and Fig. 1(c) shows the joint-axis range relative to joint-axis$_5$.

Figure 2(a) shows a fixed position of joint-axis$_1$ and a fixed position of joint-axis$_5$. For each joint axis there is a given range relative to joint-axis$_1$ and also a range relative to joint-axis$_5$. Figure 2(b) shows for each joint-axis$_j$ ($1 < j < 5$) the intersection $P_{1,j,5}$ between $R(1,j)$ and $R(5,j)$. Each intersection contains all possible joint-axis positions. Note in Fig. 2(b) that we may choose the position of joint-axis$_2$ on arc 2, joint-axis$_3$ in the black area and joint-axis$_4$ on arc 4.

When one wants to find one solution for the position of the joint axis, one can start with the intersection $P_{1,4,5}$ first (see also arc 4 of Fig. 2(b)). Figure 3(a) shows $R(1,4)$, the whole area inside the large circle, and $R(5,4)$ the small circle 4. The intersection $P_{1,4,5}$ is then the arc of the

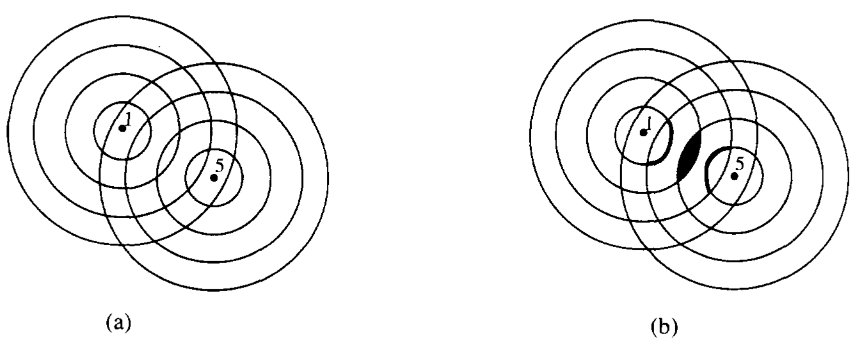



Fig. 2. Chosen position for joint 5.

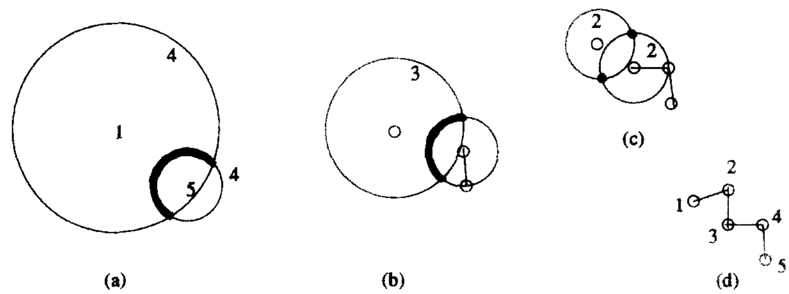



Fig. 3. Several steps in the intersection method.

small circle inside the large one, depicted as the bold arc. In formula form:

$R(1,4) := \mathrm{CI}_2(\text{position}_1, 3L)$,

$R(5,4) := C_2(\text{position}_5, L)$,

$P_{1,4,5} := R(1,4) \cap R(5,4)$.

If a position is chosen on the bold arc for joint-axis$_4$, then the algorithm is to be repeated with the reduced manipulator $R(1,4)$. So we can calculate $R(1,3)$ and $R(4,3)$ and the intersection $P_{1,3,4}$ is again an arc, see Fig. 3(b). In formula form:

$R(1,3) := \mathrm{CI}_2((x_1, y_1, z_1), 2L)$,

$R(4,3) := C_2((x_4, y_4, z_4), L)$,

$P_{1,3,4} := R(1,3) \cap R(4,3)$.

After choosing joint-axis$_3$, only the reduced manipulator $R(1,3)$ is left. Note that the intersection $P_{1,2,3}$ in this case has only two possible positions, the bold points in Fig. 3(c). In formula form:

$R(1,2) := C_2((x_1, y_1, z_1), L)$,

$R(3,2) := C_2((x_3, y_3, z_3), L)$,

$P_{1,2,3} := R(1,2) \cap R(3,2)$.

At this stage (Fig. 3(d)) all positions of the joint axis have been calculated and the joint angles can easily be derived.

### 3.2. The PUMA manipulator

The inverse kinematic problem of the PUMA manipulator (see Fig. 4(a)) looks very complex

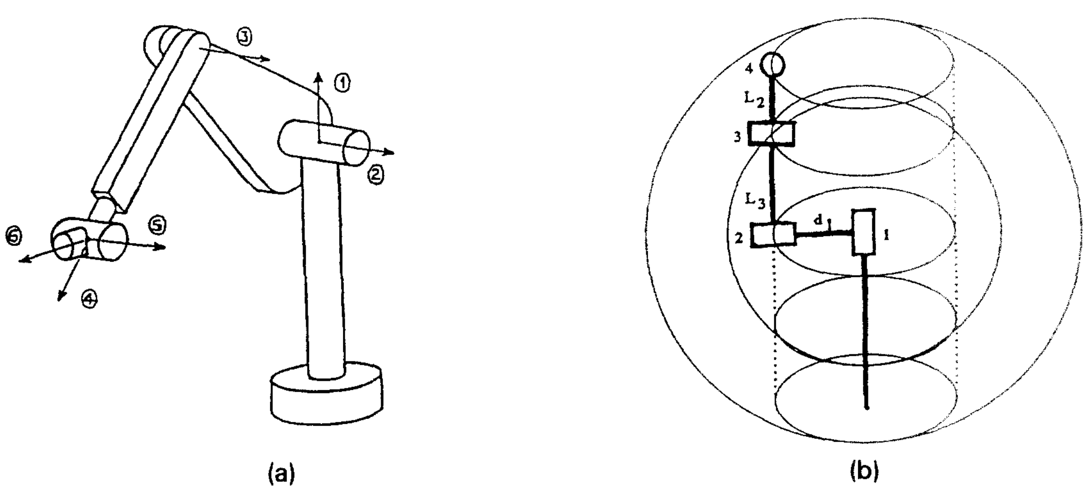



Fig. 4. The PUMA manipulator and some of its ranges.

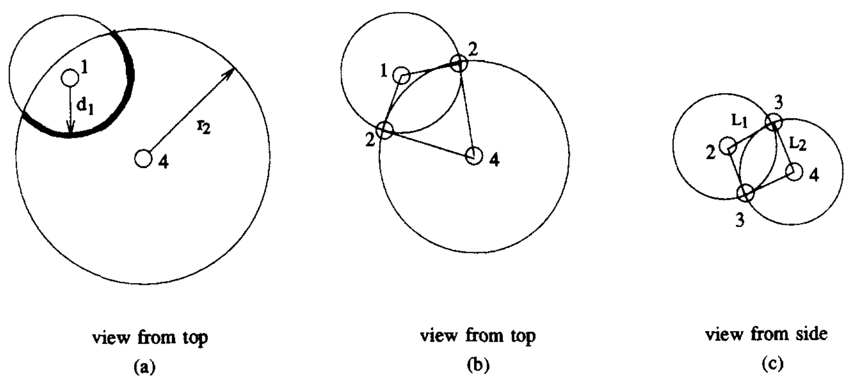



Fig. 5. Steps in the Intersection method for the PUMA manipulator.

but, as will be shown, it can be solved easily. Note that joint-axes 4, 5 and 6 form a wrist and can be considered as one spherical joint placed on the coordinates of joint-axis$_4$. Its position can be derived from the desired end effector position.

To understand more clearly the involved ranges we shall consider first ranges $R(1,j)$ where $j > 1$ (see Fig. 4(b)):

- $R(1,2)$ can be described as the circle $C_3$(position$_1$,$d_1$,(0,0,1)).
- $R(1,3)$ are all positions on the surface of the sphere $S$(position$_1$,$r_3$) with $r_3 = (d_1^2 + L_3^2)^{1/2}$, but excluding points inside a cylinder with joint-axis$_1$ as axis and a radius of $d_1$.
- $R(1,4)$ consists of all points inside the sphere $S$(position$_1$,$r_4$) with $r_4 = (d_1^2 + (L_3 + L_2)^2)^{1/2}$ and excluding the same cylinder as with $R(1,3)$.

The strength of the intersection method is that we can simply split the manipulator into two planar manipulators $R(1,2)$ and $R(4,2)$, as shown in Fig. 5(a). If there were no constraints on the direction of joint-axis$_2$, the intersection $P_{1,2,4}$ would be the entire bold arc, but if the direction of joint-axis$_2$ is considered as well, the intersection $P_{1,2,4}$ contains only two possible positions. These two positions can be easily derived by intersecting two circles (Fig. 5(b)):

$$\text{projected\_distance}(1,4) = \sqrt{(x_4 - x_1)^2 + (y_4 - y_1)^2},$$

$$r_2 = \sqrt{\text{projected\_distance}(1,4)^2 - d_1^2},$$

$$P_{1,2,4} = C_2((x_1, y_1), d_1) \cap C_2((x_4, y_4), r_2).$$

After the position of joint-axis$_2$ has been chosen, the position of joint-axis$_3$ lies on the intersection of (looking from the side, in the vertical plane through axis 2 and 4):

$$P_{1,2,4} = C_2(\text{projected\_position}_2, L_1) \cap C_2(\text{projected\_position}_4, L_2).$$

If a position has been chosen for joint-axis$_3$, all joint positions will be known. All necessary joint angles can be derived from these positions.

## 4. Conclusions

In this paper a novel approach towards the inverse kinematics problem has been presented.

Its main strong points are its intuitive clarity and wide range of applicability. It was shown that the method leads to attractively simple solutions for some widely used manipulators.

Further research is needed into the description of the joint workspaces and the calculation of their intersections. Moreover the theoretical framework of this intersection method could be more firmly established and formally specified. There still remains a lot of promising research to be done in these directions.

## References

[1] D.L. Pieper and B. Roth, "The kinematics of manipulators under computer control", *Proc. 2nd Int. Conf. on the Theory of Machines and Mechanisms*, 1969, pp. 159–168.

[2] V. Milenkovic and B. Huang, "Kinematics of major robot linkages", *Proc 13th Int. Symp. on Industrial Robots*, Chicago, IL, April 17–19, 1983, pp. 16-31–16-47.

[3] C.S.G. Lee and M. Ziegler, "A geometric approach in solving the inverse kinematics of PUMA robots", *IEEE Trans. Aerosp. Electron. Syst.*, Vol. AES-20, No. 6, 1984, pp. 87–97.

[4] G.P.A. Alards, "Two new approaches to the inverse kinematics problem", Graduation Thesis 111, Informatics Department, University of Nijmegen, The Netherlands, 1989.