DISTRIBUTED METHOD FOR INVERSE KINEMATICS OF ALL SERIAL MANIPULATORS

S. REGNIER, F.B. OUEZDOU and P. BIDAUD
Laboratoire de Robotique de Paris
Université Paris 6 - Université Versailles Saint-Quentin
CNRS URA 1778
10-12 Avenue de l'Europe
78140 Vélizy
Email: {regnier,fbo,bidaud}@robot.uvsq.fr

Abstract: This paper deals with a new numerical method to solve the inverse kinematics of all serial, special or general, manipulators. This method uses new concept from Distributed Artificial Intelligence, multi-agent systems, which allows to distribute the resolution of this problem. This concept is used with a new formulation of the problem associated to each local frame. This iterative and distributed algorithm is able to find all solutions of the inverse kinematics for all kinds of manipulators (6R, 5R1P, 4R2P, 3R3P). Moreover, we'll show that this method can be applied to redundant manipulators.

1 INTRODUCTION

In the last few years a new research paradigm has come into the international scientific area. The distributed artificial intelligence and multi-agent system has gained major importance as a paradigm for computer scientists. The word agent [1] is used to designate an intelligent entity, acting rationally and intentionally with respect to its own goals and to the current state of its knowledge. We focused on distributed problem solving where tasks are initially specified and distributed among several agents. We will take advantages of the physical distribution of the manipulators to apply distributed problem solving for the inverse kinematic problem.

This problem is to find a set of joint-variables values that will place the end effector into a given position and orientation. Our purpose is to find numerically all solutions of the inverse kinematics by a multi-agent method which could be applied to all serial manipulators. Our idea was to consider each body of the manipulator as an agent usefully cooperating towards the same collective purpose, which is reaching the position and the orientation of the end effector. Thanks to this dialogue between agents, the placing of the end effector is performed by successive stages. The main interest of this paper is a new numerical method to find all solutions for inverse kinematics of all manipulators.

The first part of this paper is devoted to the paradigm of distributed problem solving and after to a recalled of the problem of the inverse kinematic. After we'll introduce a distributed method to find **one** solution of the inverse kinematics. The fourth part concerns the generalization to find **all** solutions. Then, the results obtained by this new method are presented. In our conclusion, we have emphasized both the prospects and the limits of our method.

2 DISTRIBUTED PROBLEM SOLVING AND MULTI-AGENT SYSTEMS

Unlike the classical Artificial Intelligence (AI) which models the intelligent performances of one agent only, the Distributed Artificial Intelligence (DAI) is concerned with intelligent performances which are the product of the cooperative activity of several agents. The passage from individual performance to collective performance is considered not only as an extension but as an enrichment of (AI) as well.

An agent is an autonomous entity which pursues an individual purpose, which can act on the environment of the system. It can interact with the other agents, has only an evolutive representation of the environment and can perceive the other agents (communication, observation). The agents have two tendencies: a social and individual tendency with mechanisms and data containing the rules of the agent internal working. Then, an agent can be

characterized by its role, speciality, aim, belief, capacities of decision, of communication and possibly of training. The term multi-agents system has been applied to any system composed of multiple interacting agents.
Research in DAI has been mostly oriented towards multi-agent systems composed of sophisticated agents. These agents are categorized as cognitive agents, which are given the capabilities to reason about their environment, to predict future events and choose between possible actions, and exhibit goal-driven behavior. In the context of problem solving, this line of research is called Distributed Problem Solving (DPS).
Specially, the development of the distributed problem solving framework involves the following subject:

- *problem decomposition:* The transformation from a problem to a society of simple agents is defined by a decomposition scheme. Each agent is assigned to a task corresponding to a part of the problem.
- *behavior design:* An agent's behavior corresponds to various actions and it performs to achieve its goal.
- *coordination:* Group behavior of agents is characterized by the coordination mechanism society. For our problem-solving, we require for rapid convergence to improve problem-solving efficiency
- *system:* It follows the construction of a distributed problem solving. A society of agents are created according to specification of the problem.

So, solving complex problems is made by combining many simple solutions in an iterative manner instead of attempting to construct a single global solution. We will take advantage of the physical distribution of the manipulators to apply distributed problem solving for the inverse kinematic problem.

3 THE INVERSE KINEMATIC PROBLEM

We use classical Denavit-Hartenberg notations [17] to describe the structure of the manipulator. Each link is represented by the line along its joint axis and the common normal to the next joint axis. A coordinate system is attached to each link for describing the relative arrangements of the various links. The 4×4 transformation matrix relating j^{th} coordinate system to $(j-1)^{th}$ coordinate system is noted T_{j-1}^{j}.

$$T_{j-1}^{j} = \begin{pmatrix} \cos(\theta_j) & -\cos(\alpha_j)\sin(\theta_j) & \sin(\alpha_j)\sin(\theta_j) & a_j\cos(\theta_j) \\ \sin(\theta_j) & \cos(\alpha_j)\cos(\theta_j) & -\sin(\alpha_j)\cos(\theta_j) & a_j\sin(\theta_j) \\ 0 & \sin(\alpha_j) & \cos(\alpha_j) & d_j \\ 0 & 0 & 0 & 1 \end{pmatrix} = \left(\begin{array}{c|c} R_{j-1}^{j} & P_{j-1}^{j} \\ \hline 0 & 1 \end{array}\right) \quad (1)$$

The problem of the inverse kinematics correspond to computing the joint values θ_i or d_i $(p = 1..n)$ where n is the number of joints $(n <= 6)$ and T^h the task matrix expressed in the referential frame such that:

$$T_0^1 T_1^2 \cdots T_{n-1}^n = T^h \quad (2)$$

The methods that calculate the solutions of this are either algebraic or numerical. The first algebraic complete solution has been given by Lee and Liang [2] for general 5R1P or 6R manipulators. They calculate the 16^{th} degree characteristic polynomial in the tangent of the half angle of one of joint variables. Raghavan and Roth [3] presented the first method that may be used to calculate the characteristic polynomial of all general geometry manipulators using dyalitic elimination. Then, Manocha and Canny [4] and Cohley and Ovastic [5] involve symbolic preprocessing matrix computations and a variety of numerical techniques in order to decrease the execution time of resolution. Finally, Mavroidis and al. [6] present an efficient algorithm that solves the inverse kinematics problem of all six degrees of freedom manipulators using symbolic computing. This 16^{th} degree polynomial is prone to numerical ill-conditioning when a root yielding is an angle of π [7]. However, using an another univariable, it is possible to find an another characteristic polynomial. So, these algebraic methods are efficient except if all joint variables are equal to π or if the degree characteristic isn't minimum.
Numerical iterative schema based on Newton-Raphson has been used to obtain some of the solutions of the inverse kinematics [8]. However this method is known to have quadratic convergence in the neighborhood of a solution, but far from it either procedure can fail to converge. Angeles [9] proposed to apply small enough condition number of the Jacobian and continuation method to over-step this failure. Optimization methods [10] are also used to

solve inverse kinematic problem. They used non-linear square approximation to an over determinated algebraic system of kinematic closure equations. But theses methods can't give all solutions. Tsai and Morgan [11] applied continuation method to solve all distinct solutions for a general manipulator using six revolute joints. The continuation method requires a start system of equations that can readily be solved, a target system for which solution are sought, and a path following strategy to move from the start system to the target system. This new approach [12] is reliable on problems twist angle if at least 0.1 degree.

4 DISTRIBUTED METHOD FOR COMPUTING ONE SOLUTION OF THE INVERSE KINEMATICS

4.1 Local associated equations

The general form of inverse kinematics problem can be given in a local frame associated with the id^{th} joint (on fig. 1):

$$T_{id-1}^{id} T_{id}^{n} = T_{id-1}^{h} \tag{3}$$

with $T_{id-1}^{h} = (T_0^{id-1})^{-1} T_0^h$.

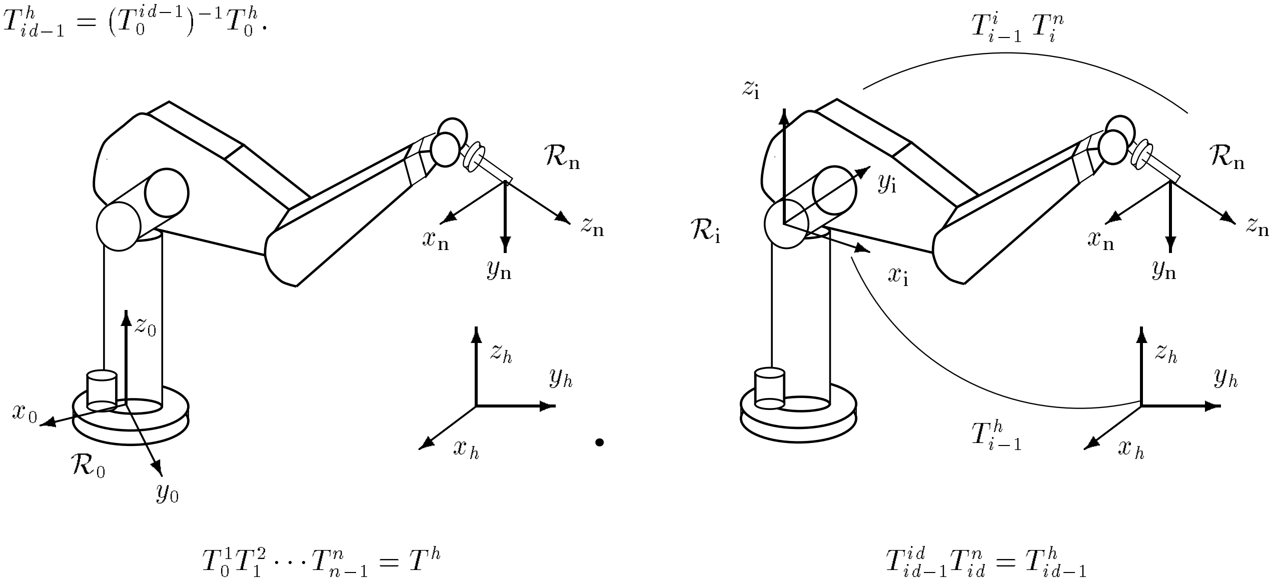


Figure 1: Inverse kinematic problem and local associated transformation

This general form can be written in different ways. If we consider that k joints between the id^{th} and the n^{th} joint are **locked** and constitute with the id^{th} link only one link then equation (3) becomes:

$$T_{id-1}^{id} T_{id}^{id+1} \cdots T_{id+k-1}^{id+k} = T_{id-1}^{h} T_n^{id+k} \tag{4}$$

So, we note:

$$T^{sup} = \prod_{p=0}^{k-1} T_{id+p}^{id+p+1} \tag{5}$$

The right term can be expressed as:

$$T^{dr} = T_{id-1}^{h} T_n^{id+k} \tag{6}$$

We have a general form of equation (4) depending on the **class** k $(0 \le k \le n-1)$ which can be written:

$$T_{id-1}^{id} T^{sup} = T^{dr} \tag{7}$$

In the following section, the problem of measuring "closeness" between frames is discussed.

4.2 Distance Metrics

Assuming an inertial reference frame and length scale for physical space have been chosen, each frame can be assigned an element of the Euclidean group $SE(3)$ (also known as the homogeneous transformations). The problem of precisely "closeness" between frames then reduces to the equivalent mathematical problem of defining a distance metrics in $SE(3)$.
Any number of arbitrary distance metrics can be defined [13] but certain features make the metrics more physically meaningfull. Since any distance metric in $SE(3)$ combines position and orientation, one would like the metric to be scale-invariant: the distance between two frames should be invariant (up to constant scaling factor) with respect to choice of length scale for physical space. Park [14] suggests a particular left-invariant distance parameterized by length scale that is useful for kinematic applications.
In this paper, we base our measure on the Frobenius norm of a matrix $\|M\|$ with a length scale L defined by Wampler [15] such that:

$$T_1 = \left(\begin{array}{c|c} R_1 & P_1 \\ \hline 0 & 1 \end{array}\right) \qquad T_2 = \left(\begin{array}{c|c} R_2 & P_2 \\ \hline 0 & 1 \end{array}\right)$$

$$\begin{aligned} d(T_1, T_2) = d((R_1, P_1), (R_2, P_2)) &= \|R_1 - R_2\|^2 + \frac{1}{L^2}\|P_1 - P_2\|^2 \qquad (8) \\ &= \|R\|^2 + \frac{1}{L^2}\|P\|^2 \\ &= \sum_{ij} R_{ij}^2 + \frac{1}{L^2}\sum_{ij} P_{ij}^2 \\ L &= \max_q \|P_0^n\| \qquad (9) \end{aligned}$$

4.3 A distributed criterion

If we consider a distributed method, just one joint is able to move at each step of our approach. The general form given by equation (7) can be expressed by a difference between the two terms of the equation:

$$M_{id} = M(q_{id}, T^{sup}, T^{dr}, a_{id}, \alpha_{id}) = T_{id-1}^{id} T^{sup} - T^{dr} \qquad (10)$$

This matrix M_{id} depends on the value of the id^{th} joint parameter (q_{id}), all geometric parameters and the task data. If we consider that only the id^{th} joint can move, we can express a value of the joint parameter that leads matrix M_{id} goes to null matrix. In this case, the left term of equation (7) tends to the right one.
Using the Frobenius norm of M_{id}, we obtain:

$$||M_{id}|| = 2\sigma_{id}(DEN_\theta * sin(\theta_{id}) + NUM_\theta * cos(\theta_{id})) + (1 - \sigma_{id})(d_{id} - d_m)^2 + CST_\theta \qquad (11)$$

with $\sigma_{id} = 1$ if the joint is revolute and $\sigma_{id} = 0$ if the joint is prismatic and DEN_θ, NUM_θ, CST_θ and d_m expressed in annex 2.
We have to consider the minimization of this norm in order to approach our goal.

$$\frac{\partial ||M_{id}||}{\partial q_{id}} = 2\sigma_{id}(-DEN_\theta * cos(\theta_{id}) + NUM_\theta * sin(\theta_{id})) + 2(1 - \sigma_{id})(d_{id} - d_m) \qquad (12)$$

In this case the expression given by equation (12) leads to the extremum values of q_{id} by writing $\frac{\partial \|M_{id}\|}{\partial q_{id}} = 0$.

$$q_{id}^m = \sigma_{id}\theta_{id}^m + (1 - \sigma_{id})d_{id}^m \qquad (13)$$

with

$$\begin{aligned} \theta_{id}^m &= \arctan\frac{NUM_\theta}{DEN_\theta} \qquad (14) \\ d_{id}^m &= d_m \qquad (15) \end{aligned}$$

It's obvious that these values give minimum value of $||M_{id}||$ due to the fact that for the revolute joint, tangent function is an increasing one and for a prismatic joint the coefficient of (d) in equation (12) is equal to 1.0 (strictly positive).

4.4 Resolution algorithm

We consider each body of the manipulator as an agent. The multi-agent system models the manipulator in this way.
Each agent id possesses the following knowledge:

- a local task matrix T^h_{id-1} which represents its local goal matrix
- its 4×4 transformation matrix T^{id}_{id-1} relating the $(id)^{th}$ joint and the $(id-1)^{th}$ joint which expresses its potential action about the world
- the 4×4 transformation matrix relating the $(id-1)^{th}$ joint and the first joint T^{id-1}_0 which represents its position in the absolute frame.

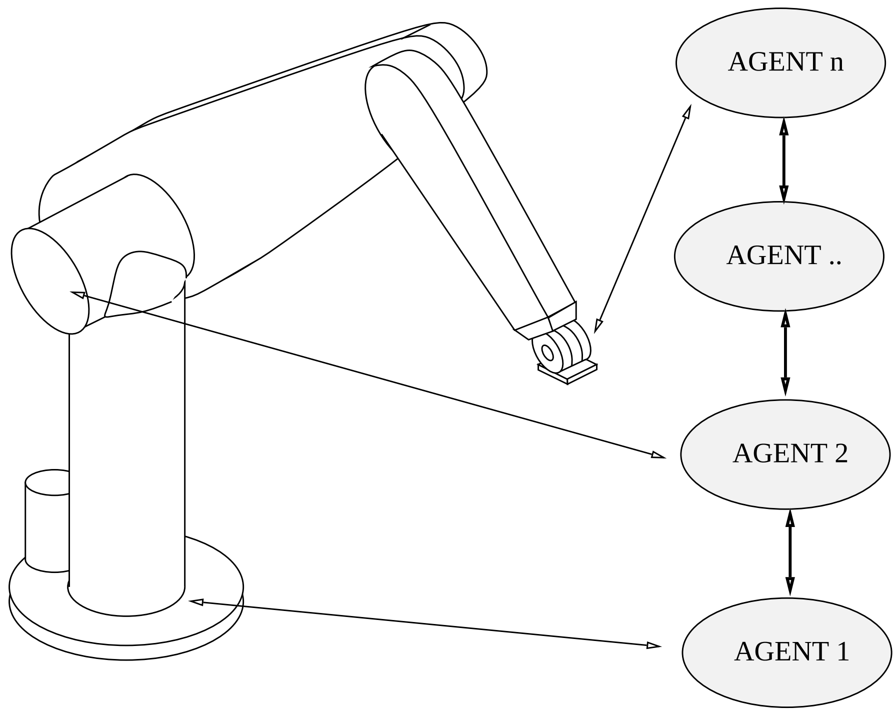


Figure 2: Manipulator and proposed modelisation by a multi-agent system

We consider a distributed paradigm of communication where the agents could only communicate with the previous or next agent (on fig. 2). Each id^{th} joint or agent must compute the value in order to minimize the norm of the matrix M_{id}. The elements of matrices T^{up} et T^{dr} depend on the others joints and task data.
So, the iterative method follows this algorithm with ϵ the accuracy of the distributed method given by the user.

Loop until $\|T_0^1 T_1^2 \cdots T_{n-1}^n - T^h\| \leq \epsilon$
    For $id = n \cdots 1$ loop
        Agent id computes (T^{dr}, T^{up})
        Agent id computes q_{id}
    End for
End loop

Figure 3: Distributed resolution algorithm

4.5 The criterion class

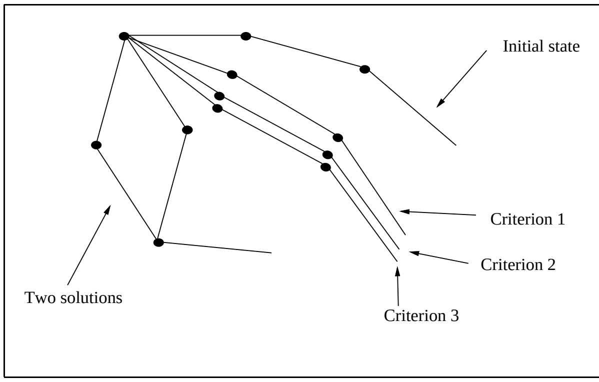
For a serial manipulator, we show that different classes of minimization criterion can be chosen. *This choice of the class is important in the algorithm* because the forms of the equation with T^{up} and T^{dr} are different. The class criterion k $(0 \leq k \leq n)$ fixes the number of joints beyond the id joint which are assumed to be locked.

In order to understand that criteria are different, we'll take a simple example with a planar manipulator formed by three revolute joints. We have three possible classes k ($k \leq 3$) of criteria for this manipulator.

Criterion	*calculus M_{id} of the joint id*		
1	$M_1 = T_0^1 - T^h T_3^1$	$M_2 = T_1^2 - T_1^h T_3^2$	$M_3 = T_2^3 - T_2^h$
2	$M_1 = T_0^1 T_1^2 - T^h T_3^2$	$M_2 = T_1^2 T_2^3 - T_2^h$	$M_3 = T_2^3 - T_2^h$
3	$M_1 = T_0^1 T_1^3 - T^h$	$M_2 = T_1^2 T_2^3 - T_2^h$	$M_1 = T_2^3 - T_2^h$

Figure 4: calculus of M_{id} for the planar manipulator

For one step of the method, we try these criteria with the first joint on fig. 5: The proposed results by the three



- Initial State: $\theta_1 = 0^o$ $\theta_2 = 20^o$ $\theta_3 = 40^o$
- Final State:
  Criterion 3: $\theta_1 = 37.87^o$ $\theta_2 = 20^o$ $\theta_3 = 40^o$
  Criterion 2: $\theta_1 = 34.45^o$ $\theta_2 = 20^o$ $\theta_3 = 40^o$
  Criterion 1: $\theta_1 = 19.77^o$ $\theta_2 = 20^o$ $\theta_3 = 40^o$
- Task matrix $T^h = \begin{bmatrix} 0.371 & 0.594 & 1.279 \\ 0.590 & 0.911 & -2.0 \\ 0.0 & 0.0 & 1.0 \end{bmatrix}$

Figure 5: Planar manipulator with three classes of criterion

criteria are different. With a serial manipulator, we have n different classes of criterion. So we'll propose to study the choice of a criterion in the next paragraph.

4.6 Choice of the criterion class

If some special conditions of the Denavit and Hartenberg parameters exist, then the manipulator is said to have a special geometry. If, in addition, these special conditions cause a manipulator to have a lower degree charecteristices polynomial than the general manipulator of the same type, we call this manipulator special manipulator otherwise the manipulator is called general manipulator [6].

Type of manipulators	3R3P	4R2P	5R1P	6R
Degree of polynomial characteristic	2	8	16	16

Figure 6: Degree characteristic polynomial for all kind of manipulators

We specified the inverse kinematic problem of a number of general and special manipulators to test the validity of our method. Neverless, we didn't find general manipulator 5R1P or 4R2P with 16 or 8 different solutions. But we introduce some problems with special manipulators (pb $n^o 4, 5, 8, 9$ in annex 3) in order to proof that our numerical method is able to solve this class of problem.
We chose ($\epsilon = 10^{-4}$) for all these tests. We focus on the criteria in order to detect the best criterion available for all kind of manipulators. The main result is to show that there is one criterion (**5**) for which the method converges more quickly. So in the following, we use the class criterion $(n-1)$ and equation (7) and equation (10) could be written as:

$$T_{id-1}^{id} T_{id}^{n} = T_{id-1}^{h} \qquad (16)$$

Type	Number	Manipulator	Degree of the polynomial
6R	1	Diestro Manipulator [16]	16
	2	21 General Manipulators[1]of [12]	16
	3	Arc Mate Manipulator [6]	16
	4	Special Manipulator	12
	5	Special Manipulator	8
5R1P	6	GP66 Manipulator [17]	16
	7	SHA Manipulator [18]	12
	8	Special Manipulator	8
4R2P	9	Special Manipulator	4
3R3P [2]	10	3-Cylindric Manipulator [19]	2

Table 1: Problem tests

$$M_{id} \;=\; T^{id}_{id-1}T^{n}_{id} \;-\; T^{h}_{id-1} \tag{17}$$

Choosing particular class needs some care in computing T^{up} and T^{dr} matrices for a given id joint. For a chosen

	Class of criterion					
Number of problem	0	1	2	3	4	5
1	1255	1220	1230	1190	1180	1150
2	994	986	955	943	934	912
3	1023	1025	1030	1007	987	980
4	1034	1029	1020	1015	1016	998
5	1345	1337	1312	1313	1297	1275
6	965	976	945	942	939	921
7	1089	1085	1064	1062	1058	1032
8	967	962	945	943	934	902
9	888	882	881	872	863	834
10	1045	1040	1031	992	991	984

Table 2: Number of steps for each first foundest solution

class (k), the condition that id should satisfy is:

$$id + k \leq n \qquad id = 1 \cdots n \tag{18}$$

If this condition is not satisfied, we have to decrease the class. So, a simple algorithm (fig. 7) allows to choose the class of criterion. The multi-agent method allows us to find **one** solution of the inverse kinematics for all tested manipulator (Problem $1 \cdots 10$) given by table 1. We are now going to explain a distributed method to solve all solutions of the inverse kinematics for all serial manipulators.

5 DISTRIBUTED METHOD FOR DETERMINATION OF ALL SOLUTIONS OF THE INVERSE KINEMATICS

5.1 Assumptions of resolution

It seems that our multi-agent method gives the same solution when the set of joints belongs to a particular generalized joint-space (GS). It's very difficult to get analytical form of this space representing the generalized

```
For id = 1 to n
        If (id + k > n) then
                k = k − 1
        end if
        Choose the criterion class
End for
```

Figure 7: Choice of the criterion class k

associated space (GAS) of a solution.
By definition, the generalized space associated (GAS) with a solution is:

Definition 1 *it exists a generalized sub-space that if an generalized point belongs to this subspace then the distributed method find always the same solution.*

So, the idea is to use a learning method which allows us to increase the GAS of a solution and to find step by step all solutions and their GAS.
We present the assumptions dealing with this method:

1. The generalized joint-space is considered to be a **discrete space**.
2. The accuracy of each id^{th} joint space between two points is Δq_{id}.
3. The joint-space of the id^{th} joint is Q_{id} including joint limits. So, the total number of discrete points is:
$$N_p^t = \text{IntegerPartOf}\left(\prod_{id=1}^{n} \frac{Q_{id}}{\Delta q_{id}}\right) \tag{19}$$
4. We note:
$$q^s = \{q_1, \cdots, q_n\} \text{ a } s^{th} \text{ solution of the inverse kinematic problem}$$
$$GAS_s = \prod_{i=1}^{n} [q_i^{min}, q_i^{max}] \quad \text{its associated generalized space}$$

5.2 Algorithm of resolution

We present a numerical algorithm in four steps.

- step 0: $s = 0$
- step 1: Choice of initial point (for example the zero point) [3]
$$q = q_{init}$$
- step 2: Apply the distributed method and find the s^{th} solution
  If the s^{th} solution is different of all others solution *then*
$$q^s \neq q^p \quad \forall p \in [1 \cdots s-1]$$
  The found solution is a **new** solution
  Build its GAS_s
$$GAS_s = \prod_{i=1}^{n} [q_i^{min}, q_i^{max}]$$

[3] This choice is not very important

$$s = s + 1$$

Else

The found solution is the same that the j^{th} solution

$$\exists j \in [1 \cdots s-1] \quad q^s = q^j \qquad \text{or} \qquad \exists j \in [1 \cdots s-1] \quad q^s \in GAS_j$$

Associated Generalized Joint-space is *increased*

$$\text{GAS}_j = \text{GAS}_j \bigcup \text{GAS}_s$$

End if

- step 3: Computing

$$N_p = \sum_{i=1}^{s} \text{IntegerPartOf}\left(\prod_{id=1}^{n} \frac{q_{id}^{max} - q_{id}^{min}}{\Delta q_{id}} \right)$$

- step 4: Return step 1 until $N_p < N_p^t$

5.3 Results

We use this distributed method with all tests problems and find **all** solutions of the inverse kinematics for these manipulators either general or special. We present the results (solutions, GAS) in the form given on figure 8 where the black rectangles represents the GAS for this solution. So for all the test problems, we obtain with an ADA

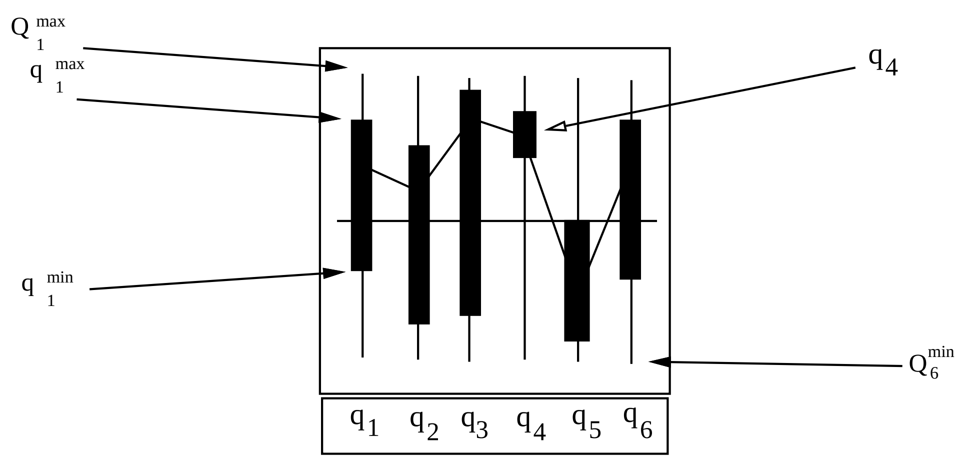


Figure 8: One solution of the inverse kinematic problem

program on PC Pentium at 75 Mhz with an accuracy $\boldsymbol{\epsilon = 10^{-4}}$ the results on figure 3.
The time of resolution is high but similar to the others numerical methods which solve only the inverse kinematics of general manipulator. We start to work to decrease it with for example centralized paradigm of communication [20] or more learning. Neverless, this method can't find all solutions in real time (similar to [4]). Its main advantages are simplicity and genericity. In fact, this method may also be applied to redundant manipulators with or without new constraints and manipulators with closed loops [21]. We'll work to analyse the articular space and singularities but it's important to understand that this distributed method is not affected by the singularities.

Pb. Number	Time	Accuracy Δq_i and joint set Q_i
Pb. N^o1	85 s	$\Delta q_1 = \cdots = \Delta q_6 = 1^o$ $Q_1 = \cdots = Q_6 = [-180^o, 180^o]$
Pb. N^o2	226 s	$\Delta q_1 = \cdots = \Delta q_6 = 1^o$ $Q_1 = \cdots = Q_6 = [-180^o, 180^o]$
Pb. N^o3	165 s	$\Delta q_1 = \cdots = \Delta q_6 = 1^o$ $Q_1 = \cdots = Q_6 = [-180^o, 180^o]$
Pb. N^o4	89 s	$\Delta q_1 = \cdots = \Delta q_6 = 1^o$ $Q_1 = \cdots = Q_6 = [-180^o, 180^o]$
Pb. N^o5	97 s	$\Delta q_1 = \cdots = \Delta q_6 = 1^o$ $Q_1 = \cdots = Q_6 = [-180^o, 180^o]$
Pb. N^o6	177 s	$\Delta q_1 = \cdots = \Delta q_6 = 1^o$ $Q_1 = \cdots = Q_6 = [-180^o, 180^o]$ $\Delta q_3 = 0.005$ $Q_3 = [0..2.0]$
Pb. N^o7	101 s	$\Delta q_1 = \cdots = \Delta q_6 = 1^o$ $Q_1 = \cdots = Q_6 = [-180^o, 180^o]$ $\Delta q_3 = 0.005$ $Q_3 = [0..3.0]$
Pb. N^o8	112 s	$\Delta q_1 = \cdots = \Delta q_6 = 1^o$ $Q_1 = \cdots = Q_6 = [-180^o, 180^o]$ $\Delta q_2 = \Delta q_5 = 0.005$ $Q_2 = Q_5 = [0..3.0]$
Pb. N^o9	74 s	$\Delta q_1 = \cdots = \Delta q_6 = 1^o$ $Q_1 = \cdots = Q_6 = [-180^o, 180^o]$ $\Delta q_2 = 0.005$ $Q_2 = [0..3.0]$
Pb. N^o10	65 s	$\Delta q_1 = \cdots = \Delta q_6 = 1^o$ $Q_1 = \cdots = Q_6 = [-180^o, 180^o]$ $\Delta q_1 = \Delta q_3 = \Delta q_5 = 0.005$ $Q_1 = Q_3 = Q_5 = [0..1.0]$

Table 3: Time of resolution and associated accuracy joint values and joint space for each problem

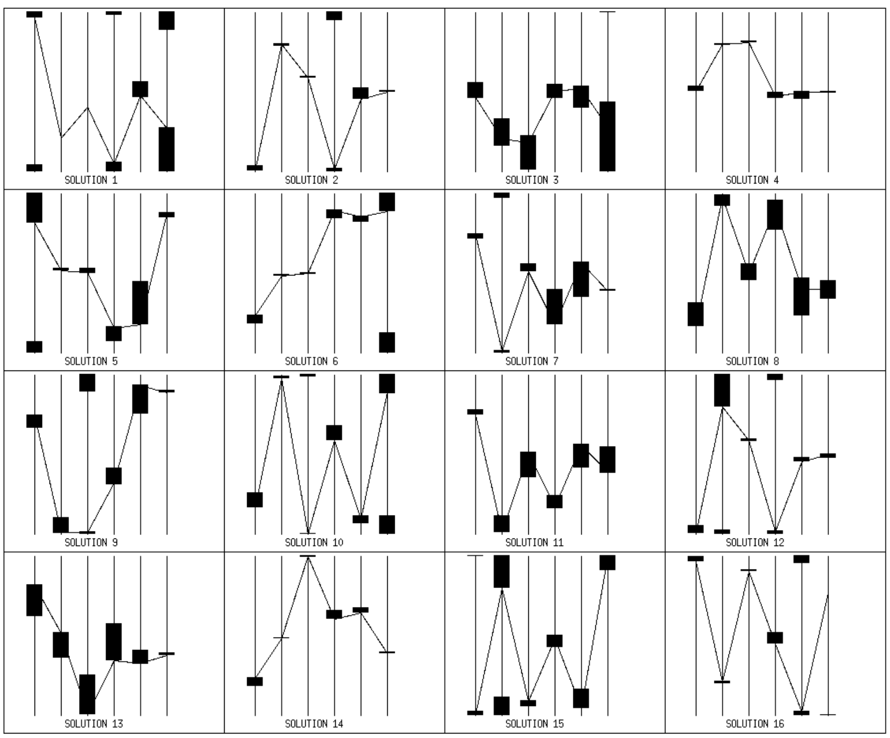


Figure 9: 16 solutions of the inverse kinematic of the problem n^o3

6 CONCLUSION

In this paper we presented a new generic and simple numerical method to solve all solutions of the inverse kinematics of all serial manipulators. This new method uses new concept multi-agent system from Distributed Artificial Intelligence to solve this problem. The idea is to associate for each local joint a new system of equations where an only joint is able to move and to approach the goal matrix. The problem of the inverse kinematics is formulated as a non-linear distributed optimization problem, in which the kinematic parameters are computed to minimize the local distance between the end-effector frame and the goal frame. This new concept is applied to solve inverse kinematics of serial manipulators and can be used to solved inverse kinematics of manipulators with closed loops [21] or forward kinematics of parallel manipulators. Morover, this distributed method can be applied with classical Denavit-Hartenberg or modified formalism and we obtain some new results to solve the kinematic design from task specification.

References

[1] B. Bond and L. Gasser, *Readings in Distributed Artificial Intelligence*. Morgan Kaufman, 1988.

[2] H. Lee and C. Liang, "A new vector theory for the analysis of spatial mechanism," *Journal of Mechanism and Machine Theory*, vol. 23, no. 3, 1988.

[3] M. Raghavan and B. Roth, "A general solution for the inverse kinematics of general 6r manipulators," in $8^{t}h$ *CISM-IFToMM Symposium on Robots and Manipulators*, pp. 24–31, 1990.

[4] D. Manocha and J. Canny, "Real-time inverse kinematics of general 6r manipulators," in *IEEE Conference Robotics and Automation*, 1992.

[5] D. Cohley and M. Ovastic, "A," *Journal of Flexible Mechanisms, Dynamics and Analysis*, vol. 47, pp. 619–629, 1992.

[6] D. Mavroidis, F. B. Ouezdou, and P. Bidaud, "Inverse kinematics of six-degree of freedom general and special manipulators using symbolic computation," *Robotica*, vol. 12, no. 5, pp. 421–430, 1994.

[7] J. Angeles and K. Zanganesh, "A semigraphical determination of all real inverse kinematics solutions of general six-revolute manipulators," in $9^{t}h$ *CISM-IFToMM Symposium on Robots and Manipulators*, 1991.

[8] K. Gupta and V. Singh, "A numerical algorithm for solving robot inverse kinematics," *Robotica*, vol. 7, pp. 159–164, 1989.

[9] J. Angeles, "On the numerical solution of the inverse kinematic problem," *The International Journal of Robotics Research*, vol. 4, no. 2, 1985.

[10] K. Kazerounian, "On the numerical inverse kinematics of robotic manipulators," *ASME Journal of Mechanisms, Transmission and Automation in Design*, vol. 109, no. 3, 1987.

[11] L. Tsai and A. Morgan, "Solving the kinematics of the most general six and five degrees of freedom manipulators by continuation method," *ASME Journal of Mechanisms, Transmission and Automation in Design*, vol. 107, 1985.

[12] C. Wampler and A. Morgan, "Solving the 6r inverse position problem using a generic case solution methodology," *Journal of Mechanism and Machine Theory*, vol. 26, no. 1, pp. 91–106, 1991.

[13] F. Park and J. Bobrow, "Efficient geometric algorithms for robot kinematic design," in *IEEE Conference on Robotics and Automation*, pp. 2132–2136, 1995.

[14] F. Park, "Distance metrics on the rigid-body motions with applications to mechanism design," *Journal of Mechanical Design*, vol. 117, no. 48, pp. 48–55, 1995.

[15] C. Wampler, "Manipulator inverse kinematic solutions based on vector formulations and damped least-squares methods," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 16, no. 1, 1984.

[16] J. Angeles and C. Lopez-Cajun, "Kinematic isotropy and the conditioning index of serial robotic manipulators," *The International Journal of Robotics Research*, vol. 11, no. 6, 1992.

[17] R. Manseur and K. Doty, "A fast algorithm for inverse kinematics analysis of robot manipulators," *The International Journal of Robotics Research*, vol. 7, no. 3, 1988.

[18] M. Kauschke and H. Rieseler, "The inverse kinematics of high solution degree manipulators," *Personal Communication*, 1992.

[19] G. Pennock and B. Viersta, "The inverse kinematics of a three-cylindric robot," *The International Journal of Robotics Research*, vol. 9, no. 4, 1990.

[20] S. Régnier, D. Duhaut, and F. Ouezdou, "A new method for the inverse kinematics," in $10^{t}h$ *CISM-IFToMM Symposium on Robots and Manipulators*, 1994.

[21] S. Régnier and D. Duhaut, "A multi-agent approach for the inverse kinematics of manipulators with closed loops," in *The Ninth World Congress on The Theory of Machines and Mechanisms*, pp. 1630–1634, 1995.

7 Annex 1

To prove that the joint value q_m leads to an accurate solution of the inverse kinematics problem, we have to consider the f^{th} step of iterative process and the joint number id. The matrix E_f represents the difference between the term of equation 2 and $\|E_f\|$ the error between the actual position of the tool and the task goal.

$$E_f = \prod_{p=1}^{n} T_{p-1}^{p} - T^h \tag{20}$$

We assume that for joint number id to n, the joint value is computed to the f^{th} step (q_p^f) and for the other joints (from 1 to $id-1$), they have the last $(f-1)^{th}$ step value (q_p^{f-1}).

$$E_f = \left\{ \prod_{p=1}^{id-1} T_{p-1}^{p}(q_p^{f-1}) \;\; T_{id-1}^{id}(q_{id}^f) \prod_{p=id+k}^{k} T_{p-1}^{p}(q_p^f) \prod_{p=k+1}^{n} T_{p-1}^{p}(q_p^{f-1}) \right\} - T^h \tag{21}$$

$$E_f = \prod_{p=1}^{id-1} T_{p-1}^{p}(q_p^{f-1}) \left\{ T_{id-1}^{id}(q_{id}^f) \prod_{p=id+k}^{k} T_{p-1}^{p}(q_p^f) \prod_{p=k+1}^{n} T_{p-1}^{p}(q_p^{f-1}) \; - T^h \right\} \tag{22}$$

$$E_f = \prod_{p=1}^{id-1} T_{p-1}^{p}(q_p^{f-1}) \left\{ T_{id-1}^{id}(q_{id}^f) \prod_{p=id+k}^{k} T_{p-1}^{p}(q_p^f) \; - T_{id-1}^{h} T_{n}^{id+k+1} \right\} T_{id+k+1}^{n} \tag{23}$$

If we note $A_{id} = \prod_{p=1}^{id-1} T_{p-1}^{p}(q_p^{f-1})$ and $B_{id} = T_{id+k+1}^{n}$ and use the definition of T^{sup} et T^{dr} matricef of equation(**??**) and (6), E_f can be written as:

$$E_f = A_{id} \left\{ T_{id-1}^{id}(q_{id}^f) \; T^{sup} \; - T^{dr} \right\} B_{id} \tag{24}$$

The Frobenius norm of E_f is:

$$\|E_f\| = \left\| A_{id} \left\{ T_{id-1}^{id}(q_{id}^f) \; T^{sup} \; - T^{dr} \right\} B_{id} \right\| \tag{25}$$

This norm can have an overvalue:

$$\|E_f\| \leq \|A_{id}\| \; \left\| T_{id-1}^{id}(q_{id}^f) \; T^{sup} \; - T^{dr} \right\| \; \|B_{id}\| \tag{26}$$

We can deduce an overvalue of $\|A_{id}\|$ and $\|B_{id}\|$:

$$\begin{aligned} \|A_{id}\| &= \left\| \prod_{p=1}^{id-1} T_{id-1}^{id}(q_{id}^f) \right\| \\ \|A_{id}\| &\leq \prod_{p=1}^{id-1} \left\| T_{id-1}^{id}(q_{id}^f) \right\| \end{aligned} \tag{27}$$

It's obvious that:

$$\|T_{p-1}^{p}(q_p)\| = 4 + a_p^2 + d_p^2 = \delta_p \tag{28}$$

So, we obtain overvalues for $\|A_{id}\|$ and $\|B_{id}\|$:

$$\|A_{id}\| \leq \prod_{p=1}^{id-1} \delta_p \tag{29}$$

$$\|B_{id}\| \leq \prod_{p=id+k+1}^{n} \delta_p \tag{30}$$

So,

$$\|E_f\| \leq |K_{id}| \; \left\| T_{id-1}^{id}(q_{id}^f) \; T^{sup} \; - T^{dr} \right\| \; \|E_f\| \leq |K_{id}| \; \|M_{id}\| \tag{31}$$

with $|K_{id}| = \prod_{p=1}^{id-1} \delta_p \; \prod_{p=id+k+1}^{n} \delta_p$.
The relation given by equation 10 can be written for each joint. So, we have an overvalue of the error $\|E_f\|$ noted Mg^f:

$$\|E_f\| \leq Mg^f \tag{32}$$

with

$$M_g^f = \sum_{p=1}^{n} |K_p| \, \|M_p^f\|$$

Or, each q_{id}^f is computed in order to minimize $\|M_{id}\|$. So, we can say that:

$$\begin{aligned} \|M_{id}(q_{id}^f)\| &\leq \|M_{id}(q_{id}^{f-1})\| \\ |K_{id}| \, \|M_{id}(q_{id}^f)\| &\leq |K_{id}| \, \|M_{id}(q_{id}^{f-1})\| \end{aligned} \tag{33}$$

So, for one step:

$$\sum_{p=1}^{n} |K_p| \, \|M_p^f\| \leq \sum_{p=1}^{n} |K_p| \, \|M_p^{f-1}\| \tag{34}$$

which implies that:

$$\|E_f\| \leq M_g^f \leq M_g^{f-1} \tag{35}$$

So at each step, the overvalue Mg^s of the error $\|E^s\|$ is decreasing and goes to zero which implies that the error it-self tends to zero. This a proof of convergence of the process to one solution of the inverse kinematics.

8 Annex 2

$$\begin{aligned}
NUM_1 &= -2\,T^n_{id1,1}T^h_{id-1_{1,1}} - 2\,T^n_{id1,4}T^h_{id-1_{1,4}} - 2\,T^n_{id1,3}T^h_{id-1_{1,3}} - 2\,T^n_{id1,2}T^h_{id-1_{1,2}} \\
&\quad -2\,aT^h_{id-1_{1,4}} + 2\,\sin(\alpha_{id})T^n_{id3,3}T^h_{id-1_{2,3}} - 2\,\cos(\alpha_{id})T^n_{id2,4}T^h_{id-1_{2,4}} \\
&\quad +2\,\sin(\alpha_{id})T^n_{id3,4}T^h_{id-1_{2,4}} + 2\,\sin(\alpha_{id})T^n_{id3,1}T^h_{id-1_{2,1}} - 2\,\cos(\alpha_{id})T^n_{id2,2}T^h_{id-1_{2,2}} \\
&\quad -2\,\cos(\alpha_{id})T^n_{id2,1}T^h_{id-1_{2,1}} - 2\,\cos(\alpha_{id})T^n_{id2,3}T^h_{id-1_{2,3}} + 2\,\sin(\alpha_{id})T^n_{id3,2}T^h_{id-1_{2,2}} \\
DEN_1 &= -2\,aT^h_{id-1_{2,4}} - 2\,\sin(\alpha_{id})T^n_{id3,4}T^h_{id-1_{1,4}} + 2\,\cos(\alpha_{id})T^n_{id2,4}T^h_{id-1_{1,4}} - 2\,T^n_{id1,3}T^h_{id-1_{2,3}} \\
&\quad -2\,T^n_{id1,4}T^h_{id-1_{2,4}} - 2\,T^n_{id1,1}T^h_{id-1_{2,1}} + 2\,\cos(\alpha_{id})T^n_{id2,3}T^h_{id-1_{1,3}} - 2\,\sin(\alpha_{id})T^n_{id3,3}T^h_{id-1_{1,3}} \\
&\quad -2\,T^n_{id1,2}T^h_{id-1_{2,2}} - 2\,\sin(\alpha_{id})T^n_{id3,1}T^h_{id-1_{1,1}} + 2\,\cos(\alpha_{id})T^n_{id2,2}T^h_{id-1_{1,2}} \\
&\quad +2\,\cos(\alpha_{id})T^n_{id2,1}T^h_{id-1_{1,1}} - 2\,\sin(\alpha_{id})T^n_{id3,2}T^h_{id-1_{1,2}} \\
\Delta_1 &= T^n_{id2,1} + T^n_{id3,1} + T^n_{id2,2} + T^n_{id3,2} + T^n_{id2,3} + T^n_{id3,3} + T^n_{id2,4} \\
&\quad +T^n_{id1,3} + a^2 + T^n_{id1,4} + T^n_{id1,2} + T^n_{id1,1} + 2T^n_{id1,4}a_d \\
r^m_{id} &= -\sin(\alpha_{id})T^n_{id2,4} - \cos(\alpha_{id})T^n_{id3,4} + T^h_{id-1_{3,4}}
\end{aligned}$$

9 Annex 3

	α_i	a_i	d_i
1	20^o	0.8	0.2
2	0^o	0.8	0.2
3	45^o	0.8	0.2
4	30^o	0.8	0.2
5	0^o	0.8	0.2
6	100^o	0.8	0.2

$$T^h = \begin{bmatrix} 0.1994 & 0.7433 & -0.6384 & 3.3051 \\ -0.9531 & 0.2984 & 0.0496 & -0.6083 \\ 0.2274 & 0.5980 & 0.7680 & 4.6235 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

θ_1	θ_2	θ_3	θ_4	θ_5	θ_6
20^o	30^o	40^o	60^o	30^o	70^o
19.296^o	-1.698^o	72.432^o	58.576^o	24.085^o	77.248^o

Table 4: Problem $n^o 4$

	α_i	a_i	d_i
1	20^o	0.8	0.2
2	30^o	1	0.7
3	45^o	0	1
4	20^o	0	0
5	30^o	0.5	0.8
6	100^o	2.2	0.6

$$T^h = \begin{bmatrix} 0.5634 & 0.6934 & -0.4491 & 4.3863 \\ -0.8226 & 0.4208 & -0.3823 & -1.5279 \\ -0.0761 & 0.5848 & 0.8075 & 2.3474 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

θ_1	θ_2	θ_3	θ_4	θ_5	θ_6
20^o	30^o	40^o	60^o	30^o	70^o
20^o	30^o	91.572^o	-60^o	147.920^o	70^o

Table 5: Problem $n^o 5$

	α_i	a_i	θ_i	d_i
1	60^o	0.7		0.8
2	90^o	0.5	30^o	
3	0^o	0.7		0.7
4	60^o	0.4		0.9
5	50^o	0.3		0.4
6	40^o	0.8		0.9

$$T^h = \begin{bmatrix} 0.649 & 0.744 & 0.155 & 3.861 \\ -0.462 & 0.550 & -0.693 & -2.369 \\ 0.602 & 0.378 & 0.703 & 0.886 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

θ_1	d_2	θ_3	θ_4	θ_5	θ_6
10^o	0.6	70^o	30^o	40^o	50^o
10^o	1.915	-70^o	170^o	40^o	50^o
2.045^o	0.503	39.266^o	57.405^o	37.523^o	57.162^o
2.045^o	1.389	-39.266^o	135.938^o	37.523^o	57.162^o
39.560^o	1.745	134.065^o	44.585^o	-39.767^o	99.071^o
39.560^o	2.751	-134.065^o	-47.283^o	-39.767^o	99.071^o

Table 6: Problem $n^o 8$

	α_i	a_i	θ_i	d_i
1	45^o	1		0.4
2	60^o	0.5	45^o	
3	0^o	0.8		0.3
4	80^o	0.7		0.5
5	30^o	0.6	45^o	
6	70^o	0.5		0.7

$$T^h = \begin{bmatrix} 0.9151 & 0.0537 & 0.3996 & 3.2785 \\ -0.0247 & 0.9967 & -0.0775 & 1.5085 \\ -0.4021 & 0.0611 & 0.9134 & 2.5806 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

θ_1	d_2	θ_3	θ_4	d_5	θ_6
20^o	0.1	50^o	10^o	0.6	70^o
30^o	0.124	44.154^o	15.845^o	0.29	70^o

Table 7: Problem $n^o 9$