Robotica (1997) volume 15, pp 653–661. Printed in the United Kingdom © 1997 Cambridge University Press

General method for kinematic synthesis of manipulators with task specifications

F.B. Ouezdou and S. Régnier

Laboratoire de Robotique de Paris, U.P.M.C.-U.V.S.Q., CNRS URA 1778, 10–12 Avenue de L'Europe, 78140 Vélizy (France) email: fbo@robot.uvsq.fr

(Received in Final Form: January 25, 1997)

SUMMARY

This paper deals with the kinematic synthesis of manipulators. A new method based on distributed solving is used to determine the dimensional parameters of a general manipulator which is able to reach a set of given tasks specified by orientation and position. First, a general *Distributed Solving Method* (*DSM*) is presented in three steps: the problem statement, the objective functions formulations and the minimum parameters values determination. Then, this method is applied to solve the synthesis of the Denavit and Hartenberg set of parameters of a manipulator with a given kinematic structure. In this case, the kind and the number of joints are specified and a set of constraints are included such as joint limits, range of dimensional parameters and geometrical obstacles avoidance. We show that if the Denavit and Hartenberg parameters (DH) are known, the synthesis problem is reduced to an inverse kinematic problem. We show also how the problem of robot base placement can be solved by the same method. A general algorithm is given for solving the synthesis problem for all kind of manipulators. The main contribution of this paper is a general method for kinematic synthesis of all kind of manipulators and some examples are presented for a six degrees of freedom manipulator in cluttered environment.

KEYWORDS: Kinematic synthesis; Manipulator; Distributed Solving Method; DH parameters.

1 INTRODUCTION

The kinematic synthesis problem can be defined as finding a set of parameters of a mechanism or a manipulator which allow it to reach a goal. This problem is known as the inverse problem by opposition the forward problem which is defined as an analysis problem. The synthesis problem was very large investigated problem since the 1970. For very simple cases, an analytical solution may be obtained. But for general 6-dof manipulators, no analytical solution exists. Many attempted to solve this problem numerically.[1–4]

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[5] 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 show[6] that this method is able to solve numerically the inverse kinematics problem of 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. Based on the same idea, we make the structural Denavit-Hartenberg parameters free to take any value included in some range, and the problem becomes now, compute all joints four DH parameters (three are constant and one variable) which satisfy the set of specified task goals. The first part of this paper is devoted for a brief summary of the Distributed Solving Method (DSM) and for a mathematical formulation of the problem including the constraints. Then the iterative process describing the method is detailed before the presentation the three kind of synthesis problems. The next part deals with solving structural parameters synthesis for an known architecture manipulator able to reach all the specified task goals with at least one solution.

In the conclusion section, we show the limits of the DSM and the feature which can be done to improve this method by taking into account other criterion such as redundancy, workspace maximization, kinematic and dynamic isotropy, etc. . .

2 DISTRIBUTED SOLVING METHOD (DSM)

The basic idea is to make the end effector of the manipulator reaching the specified task goal by minimizing the distance between a tool frame and a goal frame. So, we have to consider that each joint associated with a link is an agent able to move *doing its best* to participate for achieving this global aim. The distance between the tool and the goal frame is formulated by a Frobenius norm of a matrix representing the difference between two paths. The first path joins the base and the goal frames and the second one binds the goal frame to the base one as shown in Figure 1.

https://doi.org/10.1017/S0263574797000787 Published online by Cambridge University Press

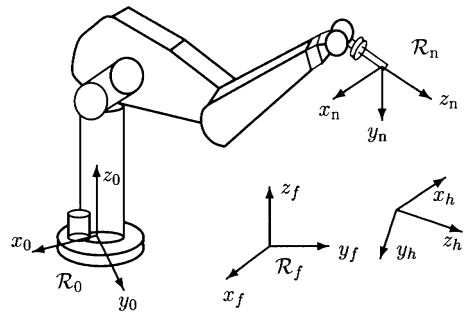


Fig. 1. Position of several frames used for the manipulator description.

2.1 Problem formulation

2.1.1 Structural and joint parameters. Using conventional Denavit and Hartenberg description of a manipulator, we have one set of constant parameters called *Structural* and another set of variable parameters (number of degrees of freedom) called *Joint* parameter. So a classical manipulator can be described by $3n$ structural parameters with n the number of joints grouped in a vector P_{dh}:

$$P_{dh} = [\alpha_1, a_1, \tilde{q}_1, \ldots, \alpha_i, a_i, \tilde{q}_i, \ldots, \alpha_n, a_n, \tilde{q}_n]^T \quad (1)$$

with $\tilde{q}_i = (1 - \sigma_i)\theta_i + \sigma_i d_i$, and $\sigma_i = 1$ for revolute joint and $\sigma_i = 0$ for a prismatic one. The end effector position and orientation in the space are given by the vector of joint parameters q:

$$q = [q_1, q_2, \ldots, q_n]^T \quad (2)$$

2.1.2 Task. The task can be specified by a six dimensions vector giving the position (X, Y, Z) and the orientation (i.e. Euler angles ϑ, φ, ψ) of the goal frame in the reference frame.

$$p = [X, Y, Z, \vartheta, \varphi, \psi]^T \quad (3)$$

2.1.3 Base frame. The position and orientation of the manipulator base frame can be specified regards to fixed global frame $\mathcal{R}_f$ with a vector p_0.

$$p_0 = [X_0, Y_0, Z_0, \vartheta_0, \varphi_0, \psi_0]^T \quad (4)$$

This allows us to solve the problem of where the base of robot should be placed to carry out the task in crowded environment[7]. So, the problem formulation gives us a $(4n + 12)$ dimensional space formed by the product of the structural $(3n)$, the joint (n), the base (6) and the task spaces (6). A manipulator or a mechanism and its instantaneous posture represent a point of this global space.

2.2. Kinematic synthesis problem

2.2.1 Kinematic structure equations. The kinematic constraints involved in the design problem can be formulated in set of kinematic structure equations which are given in a general form as following:

$$T_f^0 \overset{j}{T}_0^1 \overset{j}{T}_1^2 \cdots \overset{j}{T}_{n-1}^n = \overset{j}{T}_j^h \quad \forall j = 1 \cdots k \quad (5)$$

where $\overset{j}{T}_{i-1}^i$ the homogeneous transformation matrix (4×4) from *ith* joint frame to the $i - 1th$ one for the jth task, T_f^0 the transformation between the fixed frame to the base frame and $\overset{j}{T}_f^h$ the homogeneous transformation linking the goal frame to the fixed one. So the kinematic design problem can be addressed as computing the P_{dh} vector, the base position and orientation p_0 which allow to reach all the k tasks. So in general form, this problem can be defined as finding the solution of a set of $12 * k$ equations for $(3 + k) * n + 6$ unknowns. The set of unknown can be defined as $\mathcal{U}_t$ as:

$$\mathcal{U}_t = \bigcup_{i=1}^{n} \{\alpha_i, a_i, \tilde{q}_i\} \bigcup_{j=1}^{k} \left\{ \bigcup_{i=1}^{n} \{q_i\} \right\} \cup \{X_0, Y_0, Z_0, \vartheta_0, \varphi_0, \psi_0\} \quad (6)$$

It is obvious that, if the structural parameters are known and the base frame is defined, the set of unknowns is reduced to:

$$\mathcal{U}_t^* = \bigcup_{j=1}^{k} \left\{ \bigcup_{i=1}^{n} \{q_i\} \right\} \quad (7)$$

The problem becomes then an inverse kinematic problem of n degrees of freedom manipulator for a set of k tasks[8]. In another hand, if the base frame is defined, the number of unknowns will be $(3 + k) * n$.

2.2.2 Local frame equation. As it was said, we have to solve the structure equations (5). It is necessary to distinguish the two kind of variables: the structural variables which must be the same for all k tasks and the joint ones which depend on the task. We have to use the main advantage of the DSM by making only one joint (ith) moving at each step. All the other parameters can be considered as constant in this step. So the equation (5) can be written in a local frame associated with the *ith* joint as:

$$\overset{j}{T}_{i-1}^i \overset{j}{T}_i^n = \overset{j}{T}_{i-1}^h \quad \forall j = 1 \cdots k \quad (8)$$

with $\overset{j}{T}_{i-1}^h = (\overset{j}{T}_0^{i-1} T_f^0)^{-1} \overset{j}{T}_f^h$. The main idea of the distributed formulation is to consider at each step a sub problem of the global synthesis problem in order to get the analytical expressions of the structural and joint variables which are the best to make the *ith* joint reaching its local goal (see Figure 2).

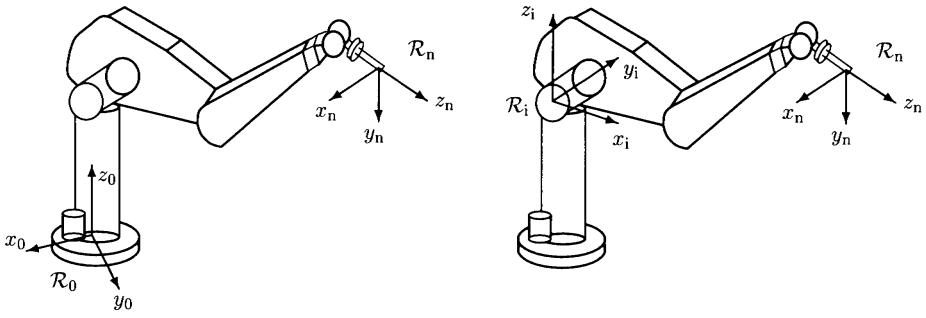
A general form of this local projection can be written as:

$$\overset{j}{\boldsymbol{T}}_{i-1}^i \overset{j}{\boldsymbol{T}}_i^n = \overset{j}{\boldsymbol{T}}_{i-1}^h \quad (9)$$

This formulation have a more meaningful sense showing that we have to compute the amount of *ith* agent contribution to solve the problem. The *ith* joint can be considered as a one degree of freedom manipulator doing its best (local) to reach the global goal.

2.2.3 Distance metrics. Assuming an inertial reference frame and length scale for physical space have been chosen, each frame can be assigned to an element of the special Euclidean group $SE(3)$. The problem of precisely "closeness" between frames is then reduced to the equivalent mathematical problem of defining a distance metrics in $SE(3)$. Any number of arbitrary distance

https://doi.org/10.1017/S0263574797000787 Published online by Cambridge University Press



$$T_f^0 T_0^1 T_1^2 \cdots T_{n-1}^n = T^h \qquad T_{i-1}^i T_i^n = T_{i-1}^h$$

Fig. 2. Local frame projection of the structure equation of a manipulator.

metrics can be defined[9] but some features make the metrics more physically meaningful. Since any distance metrics combines position and orientation, one would like the metric to be scale-invariant. So, in this paper, we base the measure on the Frobenius norm of a matrix $\|M\|$ with a length scale L defined by Wampler[10] such as:

$$T_1 = \left(\begin{array}{c|c} R_1 & U_1 \\ \hline 0 & 1 \end{array}\right) \quad T_2 = \left(\begin{array}{c|c} R_2 & U_2 \\ \hline 0 & 1 \end{array}\right)$$

$$d(T_1, T_2) = d((R_1, U_1), (R_2, U_2)) = \|R_1 - R_2\|^2 + \frac{1}{L^2}\|U_1 - U_2\|^2 \quad (10)$$

$$d(T_1, T_2) = \|R\|^2 + \frac{1}{L^2}\|U\|^2 = \sum_{ij} R_{ij}^2 + \frac{1}{L^2}\sum_{ij} U_{ij}^2$$

with L the characteristic length which is defined as the maximum Euclidean norm of absolute end effector position.

$$L = \max_q \|U_0^n\| \quad (11)$$

2.2.4. Distributed objective functions. In order to solve the kinematic synthesis problem, we have to minimize the distance between the tool frame and the task goal frame. This distance can be expressed by using a Frobenius norm of the matrix $\overset{j}{M}\overset{j}{M}_i$ which represents the difference between the current position and the desired position. Using the local projection of the structure equation (9), this matrix can be expressed as:

$$\overset{j}{M}\overset{j}{M}_i = \overset{j}{T}{}_{i-1}^{i}\overset{j}{T}_i^n - \overset{j}{T}_{i-1}^h = F(\overset{j}{F}, \overset{j}{T}_i^n, \overset{j}{T}_{i-1}^h, a_i, \alpha_i, \tilde{q}_i) \quad (12)$$

So we have to determine k Frobenius norms:

$$\|\overset{j}{M}_i\|^2 = \|\overset{j}{T}{}_{i-1}^{i}\overset{j}{T}_i^n - \overset{j}{T}_{i-1}^h\|^2 \quad \forall j = 1 \cdots k \quad (13)$$

For each task, the joint variables must be different, so the objective function should be:

$$\overset{j}{F}_i(\overset{j}{q}_i) = \|\overset{j}{M}_i\|^2 \quad (14)$$

The general form of the $\|\overset{j}{M}_i\|^2$ is given in Appendix A. The structural parameters must be the same for the k tasks, so each squared Frobenius norm should be equal to zero and their sum also. The objective function depending on the structural parameters (a_i, α_i and $\tilde{q}_i$) can be written as:

$$G_i^k(a_i, \alpha_i, \tilde{q}_i) = \sum_{j=1}^{k} \|\overset{j}{T}{}_{i-1}^{i}\overset{j}{T}_i^n - \overset{j}{T}_{i-1}^h\|^2 \quad (15)$$

The base positioning problem can be addressed as where we have to place the robot base frame $\mathcal{R}_0$ in the fixed frame $\mathcal{R}_f$ to be able to reach all tasks. Using the position X_0, Y_0, Z_0 and the orientation ϑ_o, φ_0, ψ_0 parameters, we define the following structure equation including T_f^0 transformation.

$$T_f^0 \overset{j}{T}_0^n = \overset{j}{T}_f^h \qquad T_f^0 = \overset{j}{T}{}^g \quad (16)$$

with $\overset{j}{T}{}^g = \overset{j}{T}_f^h(\overset{j}{T}_0^n)^{-1}$. We define the matrix $\overset{j}{M}_0$ which should be equal to zero as:

$$\overset{j}{M}_0 = T_f^0 - \overset{j}{T}{}^g \quad (17)$$

The Frobenius norm $\|\overset{j}{M}_0\|^2$ is given in Appendix A. This objective function can be defined as a function of (X_0, Y_0, Z_0) and (ϑ_0, φ_0, ψ_0) as follows:

$$H^k(X_0, Y_0, Z_0, \vartheta_0, \varphi_0, \psi_0) = \sum_{j=1}^{k} \|\overset{j}{M}_0\|^2 \quad (18)$$

2.2.5. Minimum values of parameters. Solving the kinematic synthesis problem becomes determining the values of the structural and joint parameters which minimize the global objective functions given below. So, we have now to compute the analytical expressions of the parameters giving a minimum value of each objective function.

(i) Joint parameter

The expression of the objective function F_i can be written by omitting the task index j as;

$$F_i = 2\sigma_i(DEN_\theta * \sin(\theta_i) + NUM_\theta * \cos(\theta_i)) + (1 - \sigma_i)(d_i - d_m)^2 + \Delta_\theta \quad (19)$$

The expressions of DEN_θ, NUM_θ, d_m et Δ_θ are also given in Appendix A. If only the *ith* joint can move, the

https://doi.org/10.1017/S0263574797000787 Published online by Cambridge University Press

minimum value of the joint parameter satisfies $\frac{\partial F_i}{\partial q_i} = 0$. We can compute the derivatives from equation (19) and we obtain:

$$\frac{\partial \|M_i\|^2}{\partial q_i} = 2\sigma_i(-DEN_\theta * \cos(\theta_i) + NUM_\theta * \sin(\theta_i)) + 2(1-\sigma_i)(d_i - d_m) \quad (20)$$

We get the value of the joint parameter q_i^m minimizing the squared norm F_i:

$$q_i^m = \sigma_i \theta_i^m + (1-\sigma_i) d_i^m \quad (21)$$

with

$$\theta_i^m = \arctan \frac{NUM_\theta}{DEN_\theta} \quad (22)$$

$$d_i^m = d_m \quad (23)$$

if the joint is a prismatic one, the value d_m is a minimum of the norm of M_i because its coefficient in equation (20) is equal to 1.0. However, in the case of revolute joint, there are two solutions for $\frac{\partial \|M_i\|^2}{\partial q_i} = 0$ which are θ_i^m and $\theta_i^m + \pi$. If we call $S^+ = \sin(\theta_i^m)$ and $C^+ = \cos(\theta_i^m)$, the objective function has two possible forms:

$$F_i(\theta_i^m) = S^+ DEN_\theta + C^+ NUM_\theta + \Delta_\theta \quad (24)$$

$$F_i(\theta_i^m + \pi) = -S^+ DEN_\theta - C^+ NUM_\theta + \Delta_\theta \quad (25)$$

In the range $\left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$, C^+ is positive and the value of the joint parameter giving the minimum is obtained when DEN_θ has the same sign as S^+. So we have the following condition:

$$\text{if } DEN_\theta < 0 \quad \text{then} \quad \theta_i^m = \theta_i^m + \pi \quad (26)$$

(ii) Structural parameters

The expression of the objective function G_i^k can be written in factorized form with α_i and a_i parameters:

$$G_i^k = \sum_{j=1}^{k} ((DEN_\alpha^j * \sin(\alpha_i) - NUM_\alpha^j * \cos(\alpha_i)) + (a_i^j - a_m^j)^2 + \Delta_\alpha^j) \quad (27)$$

The expressions for terms DEN_α^j, NUM_α^j, Δ_α^j and a_m^j are given in Appendix A. We have the analytical expressions of parameters values minimizing the norm G_i^k:

$$\frac{\partial G_i^k}{\partial \alpha_i} = \sum_{j=1}^{k} (-DEN_\alpha^j * \cos(\alpha_i) + NUM_\alpha^j * \sin(\alpha_i)) \quad (28)$$

which gives α_i^m

$$\alpha_i^m = \arctan \frac{\sum_{j=1}^{k} NUM_\alpha^j}{\sum_{j=1}^{k} DEN_\alpha^j} \quad (29)$$

For a_i, we can write:

$$\frac{\partial G_i^k}{\partial a_i} = a_i - a_m \quad (30)$$

and

$$a_i^m = \frac{1}{k} \sum_{j=1}^{k} a_m^j \quad (31)$$

We can see that a_i^m is the average of the solutions for each task a_m^j.

For the last parameter $\tilde{q}_i$, we can write the same norm and obtain:

$$\tilde{q}_i^m = (1-\sigma_i)\theta_i^m + \sigma_i d_i^m \quad (32)$$

with

$$\theta_i^m = \arctan \frac{\sum_{j=1}^{k} NUM_\theta^j}{\sum_{j=1}^{k} DEN_\theta^j} \quad (33)$$

$$d_i^m = \frac{1}{k} \sum_{j=1}^{k} d_m^j \quad (34)$$

Values of NUM_θ^j, DEN_θ^j and d_m^j are respectively that of NUM_θ, DEN_θ and d_m by adding the task index j. We should notice that this formulation is a general one and independent of the number of tasks which is a powerful aspect of this kinematic synthesis method.

(iii) Base frame parameters

The value of the parameters minimizing the Frobenius norm are given by solving the partial derivative of H^k through the position and orientation parameters. We obtain the following expressions of all parameters:

$$\vartheta_0^m = \arctan \frac{\sum_{j=1}^{k} NUM_{\vartheta_0}^j}{\sum_{j=1}^{k} DEN_{\vartheta_0}^j} \quad (35)$$

$$\varphi_0^m = \arctan \frac{\sum_{j=1}^{k} NUM_{\varphi_0}^j}{\sum_{j=1}^{k} DEN_{\varphi_0}^j} \quad (36)$$

$$\psi_0^m = \arctan \frac{\sum_{j=1}^{k} NUM_{\psi_0}^j}{\sum_{j=1}^{k} DEN_{\psi_0}^j} \quad (37)$$

The position vector (X_0, Y_0, Z_0) is given by

$$X_0^m = \frac{1}{k} \sum_{j=1}^{k} {}^j T^g{}_{1,4}$$

$$Y_0^m = \frac{1}{k} \sum_{j=1}^{k} {}^j T^g{}_{2,4}$$

$$Z_0^m = \frac{1}{k} \sum_{j=1}^{k} {}^j T^g{}_{3,4}$$

with the expressions of $NUM_{\vartheta_0}^j$, $DEN_{\vartheta_0}^j$, $NUM_{\varphi_0}^j$, $DEN_{\varphi_0}^j$, $NUM_{\psi_0}^j$ and $DEN_{\varphi_0}^j$ are given in Appendix A and ${}^j T^g{}_{l,m}$ is the ${}^j T^g(l, m)$ element.

2.3 Variable constraints

In several kinematic synthesis problems, we have to take into account some constraints on the variables. These can be formulated with some penalty functions. We show here how we can solve the problem of joint limits, obstacle avoidance by including penalty functions.

2.3.1. Variables range. The structural and joint variables can be limited in a such domain. The joint variables

https://doi.org/10.1017/S0263574797000787 Published online by Cambridge University Press

should satisfy joint end stops for instance. Some length must be included between a minimum value and a maximum one and so on. Assuming that each space is a discrete one, a general formulation can be written for the two kind of variables.

Angular variable $\beta \in \{\vartheta_0, \varphi_0, \psi_0\} \bigcup_{i=1}^{n} \{\alpha_i, \theta_i\}$

$$^{\min}\beta \le \beta \le {}^{\max}\beta \Rightarrow F_l(\beta) = \left\{\max\left[0, \tan\left(\frac{^{\min}\beta}{2}\right) - \tan\left(\frac{\beta}{2}\right)\right]^2 + \max\left[0, \tan\left(\frac{\beta}{2}\right) - \tan\left(\frac{^{\max}\beta}{2}\right)\right]^2\right\} \quad (38)$$

and linear $\lambda \in \{X_0, Y_0, Z_0\} \bigcup_{i=1}^{n} \{a_i, d_i\}$ as:

$$^{\min}\lambda \le \lambda \le {}^{\max}\lambda \Rightarrow F_l(\lambda) = \{\max[0, {}^{\min}\lambda - \lambda]^2 + \max[0, \lambda - {}^{\max}\lambda]^2\} \quad (39)$$

2.3.2. Complex environment. The manipulator should have a free collision trajectory in its environment. We show here that, by using the same formulation that of variable range, we can include the obstacles with a penalty functions in the distributed optimization process.

(i) Obstacle modelisation

In order, to show how this kind of problems can be solved, we use a simplified models of obstacles: each obstacle is defined as a box. The cartesian coordinates of point (X, Y, Z) which belongs to the obstacle are defined by:

$$^{\min}X_{\text{obj}} \le X_{\text{obj}} \le {}^{\max}X_{\text{obj}}$$

$$^{\min}Y_{\text{obj}} \le Y_{\text{obj}} \le {}^{\max}Y_{\text{obj}}$$

$$^{\min}Z_{\text{obj}} \le Z_{\text{obj}} \le {}^{\max}Z_{\text{obj}}$$

The manipulator is assumed to be formed by a set of thickness lines. This kind of assumptions for obstacles and manipulator are not restrictive because a real obstacle can be always approximated by a union of box primitives. The thickness of the manipulator links can be corrected by increasing the box obstacles dimensions.

(ii) Penalty function (Figure 3)

The penalty function should be proportional to the constraint violation. The manipulator has to avoid passing through the obstacle. We define a function which is equal to 1 if the manipulator intersect with the obstacle and zero other wise. The intersection length $\mathscr{L}$ will be integral over the manipulator length of this function.

Minimizing the penalty function will reduce the intersection length $\mathscr{L}$. For the two kind of variables λ_i and β_i, we can denfie the penalty function taking into account the last computed values ${}^c\lambda_i$ or ${}^c\beta_i$ as follows:

$$F_{\text{obst}} = \frac{\mathscr{L}}{L}(F_i(\lambda_i) - F_i({}^c\lambda_i))^2 \quad (40)$$

$$F_{\text{obst}} = \frac{\mathscr{L}}{L}\left(F_i\left(\tan\left(\frac{\beta_i}{2}\right)\right) - F_i\left(\tan\left(\frac{^c\beta_i}{2}\right)\right)\right)^2 \quad (41)$$

with F_i the objective function defined without obstacles and L the characteristic length of the manipulator (equation 11).

So, the objective function expression for the *ith* joint is:

$$F_i^t = F_i + F_{\text{obst}} \quad (42)$$

3 GENERAL ALGORITHM

We show that the main assumption of the DSM, is that at each step, only one variable is optimized. So a general algorithm should use this formulation by changing the whole problem of kinematic synthesis which is a multi-variables optimization to a multi-univariable optimization. The method was implemented with Ada programming language using the multi-task concept. The Ada Language offers facilities to model tasks. Tasks are entities which execution proceed in parallel in the following sense. Different tasks proceed independently, except at points where they synchronize. The general algorithm is given by Figure 4:

4 RESULTS

To illustrate the DSM capabilities, a kinematic synthesis of a general 6R manipulator with obstacle avoidance and a robot base placement problems are treated.

4.1. Example 1: Kinematic synthesis with obstacle avoidance

In this case, we have to find the best dimensional parameters of a 6R manipulator achieving a prescribed trajectory in a cluttered environment with its base fixed at the reference frame ($\mathscr{R}_0 \equiv \mathscr{R}_f$). The trajectory is defined by three parts and a total number of $k = 200$ points. At each point, both position and orientation of the manipulator end effector have to reach the desired values given on Table I.

To make the problem more complex, the prescribed trajectory is defined in a cluttered environment

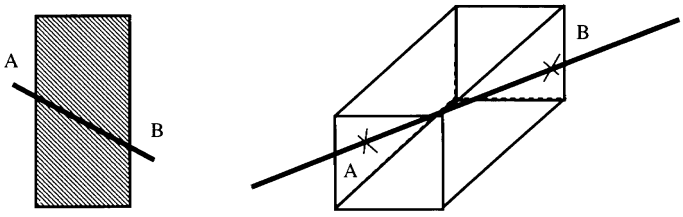


Fig. 3. Penalty function for obstacles

```
While F(P_dh, q) ≤ ε
   For i = 1 ··· n do
      For j = 1 ··· k do
         compute objective function F_i^j
         compute q_i^j
      end for
      compute P_dh(i)
   end for
end while
```

Fig. 4. General algorithm.

https://doi.org/10.1017/S0263574797000787 Published online by Cambridge University Press

Table I. Prescribed trajectory defined by 200 points

Part	Range	Desired Position (X_i, Y_i, Z_i) and Orientation (ϑ_i, φ_i, ψ_i)		
1	$0 \le i \le 89$	$X_i = 1.87 - 0.033 \times i$	$Y_i = 0.26 + 0.01 \times i$	$Z_i = 1.86 - 0.015 \times i$
		$\vartheta_i = 0 + 0.5^o \times i$	$\varphi_i = 0 + 0.5^o \times i$	$\psi_i = 0.0 + 0.5^o \times i$
2	$0 \le i \le 19$	$X_i = -1.18 + 0.001 \times i$	$Y_i = 1.18 - 0.001 \times i$	$Z_i = 0.52 + 0.032 \times i$
			$\vartheta_i = 45°$ $\varphi_i = 45°$ $\psi_i = 45°$	
3	$0 \le i \le 89$	$X_i = -1.20 + 0.033 \times i$	$Y_i = 1.16 - 0.01 \times i$	$Z_i = 1.16 + 0.01 \times i$
		$\vartheta_i = 45^o - 0.5^o \times i$	$\varphi_i = 45^o - 0.5^o \times i$	$\psi_i = 45^o - 0.5^o \times i$

Table II. Dimensions and position of obstacles

Obstacle	Length	Width	Height	X	Y	Z
A	0.5	0.5	0.8	−0.7	0	0.6
B	1	1	0.4	1.2	0	0.3
C	0.1	0.1	1.6	0.5	0	0.8
D	0.1	0.1	1.6	−1.5	0	0.8
E	0.5	0.1	2.2	1.8	1	0.2

composed by five obstacles given by their dimensions and position in the fixed frame on Table II.

A CAD representation made by ACT* software of the obstacles and the trajectory is given by Figure 5. Since the base parameters are fixed to null values, the total number of unknowns is $(3 + k) \times 6 = 1218$ which is quite large and prove the power of the DSM to solve this kind of problems. The algorithm is initialed with null values for all unknowns. The dimensional DH parameters solution of the problem are given on Table III.

The algorithm takes about 12 hours on Sparc 5 station to find the solution. The computation time is long but the problem is quite constrained: 5 obstacles, 200 tasks and the initial guess is null values. It is obvious that the results for the values of angles or distances should be changed to some ''realizable'' values. The new DH parameters are given by Table IV. We can see that the three last axis are orthogonal (i.e. twist angles $\alpha_i = \frac{\pi}{2}$, $i = 4, 5, 6$) but they do not intersect at the same point (i.e. distances d_i and a_i are different from zero). The trajectory is checked again with this new manipulator. A CAD representation of the ''realizable'' manipulator is given by Figure 6.

4.2. Example 2: base placement

In this case, we have to find the base frame parameters (X_0, Y_0, Z_0, ϑ, φ_0, ψ_0) for a given manipulator in order to reach a given set of tasks. So, we use the new realizable DH parameters values given by Table IV and re-compute the new position and orientation of the base of the 6R realizable manipulator in order to reach the same trajectory described below (Table I). The results show that it is necessary to make ''small'' change in the position and the orientation to take into account the ''small'' modifications of the initial DH parameters (Table 3). The results are given by Table V.

5 DISCUSSION

Due to the fact that at each step of our algorithm, only one parameter is modified, the DSM is less optimal than

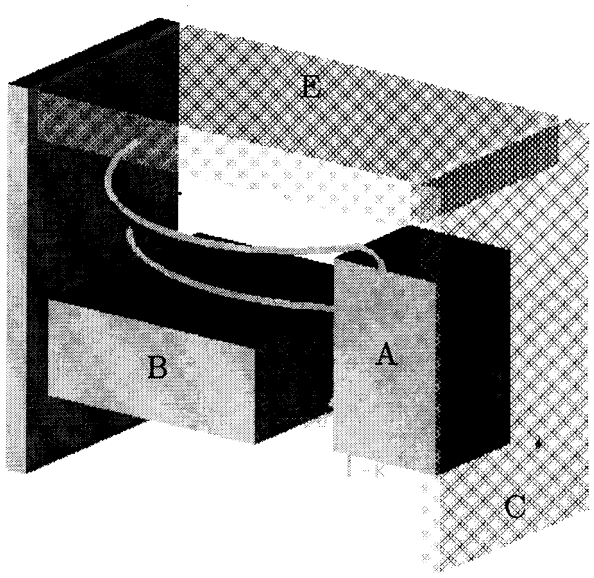


Fig. 5. CAD representation of the obstacles and trajectory.

usual numerical algorithms from computation time point of view. Using the MIPS (Million Instructions Per Second) ratio, a comparison with last works dealing with the Kinematic Synthesis Problem (KSP)[4,11] and the Robot Base Placement (RBP)[2] was carried out. The results given on Table VI, show that the proposed method is less efficient but the computation times taken are not so different. On the other hand, the DSM is a general method able to solve in the *same way* a several problems: the inverse kinematic problem, the kinematic

Table III. DH parameters of 6R manipulator

α	a	d
−17.49	0.86	1.01
−140.21	0.01	0.94
159.85	0.99	0.01
89.67	0.01	0.05
90.56	0.01	0.01
89.63	0.0	0.0

Table IV. Realizable values of DH parameters

α	a	d
17.0	0.86	1.01
−140.0	0.01	0.94
160.0	0.99	0.01
90.0	0.01	0.05
90.0	0.01	0.01
90.0	0.0	0.0

* ACT is a Trade Mark of Aleph Technologies.

https://doi.org/10.1017/S0263574797000787 Published online by Cambridge University Press

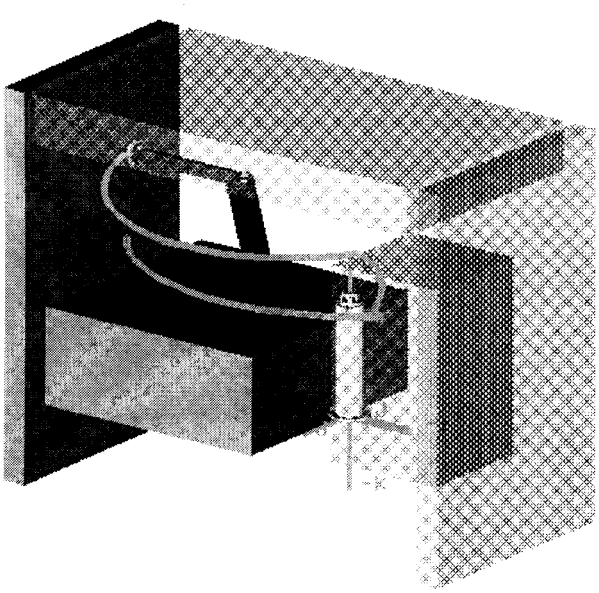


Fig. 6. Kinematic synthesis of 6R manipulator in crowded environment.

synthesis problem and the robot base placement problem. The DSM is also independent from the number of tasks and does not need a feasible starting guess of the initial values of parameters.

6 CONCLUSION

In this paper, a general method for the kinematic synthesis of general manipulators is presented. The method is based on a Distributed Solving Algorithm using a multi-agent concept. A general formulation of the kinematic synthesis problem allows us to solve several problems: the inverse kinematic problem, the determination of the structural parameters of a manipulator which is able to reach an unlimited set of positions and orientations and the robot base placement problem in cluttered environment. At each step of the algorithm, only one variable is optimized which leads us to get the analytical expression of the optimal value of this parameter. This method was tested for the design of a general 6R manipulator achieving a prescribed trajectory, and for a robot base placement problem. The method is less efficient than usual numerical algorithms. However, it is independent of the number of tasks and does not need a feasible initial guess. Further developments will concern the improvement of the method by using better initial values and including some optimization criteria to obtain an optimal solution.

Table V. Base position and orientation

X_0	Y_0	Z_0	ϑ_0	φ_0	ψ_0
−0.0124	0.0137	−0.0096	0.89^o	-1.02^o	-1.27^o

Table VI. Computation time comparisons

KSP	Park[4]: 17′20″	DSM: 19′43″
	Paredis & Khosla[11]: 12′07″	DSM: 11′45″
RBP	Chedmail & Wenger[2]: 1′30″	DSM: 2′07″

References

1. T. Subbian and D.R. Flugrad, "Use of continuation methods for kinematic synthesis" *The Eight World Congress on The Theory of Machines and Mechanisms* (1991) pp. 85–89.
2. P. Chedmail and Ph. Wenger, "Design and positioning of a robot in an environment with obstacles using optimal research" *IEEE Conference on Robotics and Automation* (1989) pp. 1069–1074.
3. A.G. Erdman and G.N. Sandor, *Mechanism Design: Analysis and Synthesis* (Prentice-Hall International Editions, USA, 1991).
4. C. Paredis and P. Khosla, "Kinematic design of serial link manipulators from task specifications" *Int. J. Robotics Research* **3**(12) 274–287 (1993).
5. B. Bond and L. Gasser, *Readings in Distributed Artificial Intelligence* (Morgan Kaufman, USA, 1988).
6. S. Régnier, C. Vougny, F.B. Ouezdou and D. Duhaut, "A new method for the inverse kinematics" *10th CISM-IFToMM Symposium on Robots and Manipulators* (1994) pp 33–38.
7. S. Zeghloul and A. Pamanes-Garcia, "Multi-criteria optimal placement of robots in constrained environments" *Robotica,* **11**, part 2, 105–110 (1993).
8. S. Régnier, "*Une méthodologie distribuées et interactive pour l'aide à la conception de systèmes robotiques*". *PhD thesis* (Université Pierre et Marie Curie, 1996).
9. F. Park and J. Bobrow, "Efficient geometric algorithms for robot kinematic design" *IEEE Conference on Robotics and Automation* (1995) pp. 2132–2136.
10. C. Wampler, "Manipulator inverse kinematic solutions based on vector formualtions and damped least-squares methods" *IEEE Transactions on Systems, Man, and Cybernetics* **16**(1), 93–101 (1986).
11. F.C. Park, "Distance metrics on the rigid-body motions with applications to mechanism design," *J. Mechanical Design,* **117**(48) 48–55 (1995).

A.1 Norm $\|M_i\|^2$

$$
\begin{aligned}
\|M_i\|^2 = {} & (\cos(\theta_i)T^n_{i\ 1,1} - \sin(\theta_i)\cos(\alpha_i)T^n_{i\ 2,1} \\
& + \sin(\theta_i)\sin(\alpha_i)T^n_{i\ 3,1} - T^h_{i-1\ 1,1})^2 \\
& + (\cos(\theta_i)T^n_{i\ 1,2} - \sin(\theta_i)\cos(\alpha_i)T^n_{i\ 2,2} \\
& + \sin(\theta_i)\sin(\alpha_i)T^n_{i\ 3,2} - T^h_{i-1\ 1,2})^2 \\
& + (\cos(\theta_i)T^n_{i\ 1,3} - \sin(\theta_i)\cos(\alpha_i)T^n_{i\ 2,3} \\
& + \sin(\theta_i)\sin(\alpha_i)T^n_{i\ 3,3} - T^j_{i-1\ 1,3})^2 \\
& + (\cos(\theta_i)T^n_{i\ 1,4} - \sin(\theta_i)\cos(\alpha_i)T^n_{i\ 2,4} \\
& + \sin(\theta_i)\sin(\alpha_i)T^n_{i\ 3,4} + a_i\cos(\theta_i) - T^h_{i-1\ 1,4})^2 \\
& + (\sin(\theta_i)T^n_{i\ 1,1} + \cos(\theta_i)\cos(\alpha_i)T^n_{i\ 2,1} \\
& - \cos(\theta_i)\sin(\alpha_i)T^n_{i\ 3,1} - T^h_{i-1\ 2,1})^2 \\
& + (\sin(\theta_i)T^n_{i\ 1,2} + \cos(\theta_i)\cos(\alpha_i)T^n_{i\ 2,2} \\
& - \cos(\theta_i)\sin(\alpha_i)T^n_{i\ 3,2} - T^h_{i-1\ 2,2})^2 \\
& + (\sin(\theta_i)T^n_{i\ 1,3} + \cos(\theta_i)\cos(\alpha_i)T^n_{i\ 2,3} \\
& - \cos(\theta_i)\sin(\alpha_i)T^n_{i\ 3,3} - T^h_{i-1\ 2,3})^2 \\
& + (\sin(\theta_i)T^n_{i\ 1,4} + \cos(\theta_i)\cos(\alpha_i)T^n_{i\ 2,4} \\
& - \cos(\theta_i)\sin(\alpha_i)T^n_{i\ 3,4} + a_i\sin(\theta_i) - T^h_{i-1\ 2,4})^2 \\
& + (\sin(\alpha_i)T^n_{i\ 2,1} + \cos(\alpha_i)T^n_{i\ 3,1} - T^h_{i-1\ 3,1})^2 \\
& + (\sin(\alpha_i)T^n_{i\ 2,2} + \cos(\alpha_i)T^n_{i\ 3,2} - T^h_{i-1\ 3,2})^2 \\
& + (\sin(\alpha_i)T^n_{i\ 2,3} + \cos(\alpha_i)T^n_{i\ 3,3} - T^h_{i-1\ 3,3})^2 \\
& + (\sin(\alpha_i)T^n_{i\ 2,4} + \cos(\alpha_i)T^n_{i\ 3,4} + d_i - T^h_{i-1\ 3,4})^2
\end{aligned}
$$

https://doi.org/10.1017/S0263574797000787 Published online by Cambridge University Press

A.2 norm $\|M_0\|^2$

$$\begin{aligned}\|M_0\|^2 = {} & X_0^2 + Y_0^2 + Z_0^2 - 2\cos(\vartheta_0)\cos(\psi_0)T^g{}_{2,2} \\ & + T^g{}_{3,2}{}^2 + 2\sin(\varphi_0)T^q{}_{3,1} + T^g{}_{3,3}{}^2 \\ & - 2Z_0 T^g{}_{3,4} + T^g{}_{3,4}{}^2 + T^g{}_{1,4}{}^2 + T^g{}_{3,1}{}^2 \\ & + T^g{}_{2,4}{}^2 + T^g{}_{2,2}{}^2 + T^g{}_{2,3}{}^2 \\ & + T^g{}_{1,2}{}^2 + T^g{}_{1,3}{}^2 - 2X_0 T^g{}_{1,4} + T^g{}_{2,1}{}^2 \\ & + 2\sin(\vartheta_0)\cos(\psi_0)T^g{}_{1,2} \\ & - 2\sin(\vartheta_0)\sin(\psi_0)T^q{}_{1,3} - 2\sin(\vartheta_0)\cos(\varphi_0)T^g{}_{2,1} \\ & - 2\cos(\varphi_0)\cos(\psi_0)T^g{}_{3,3} \\ & + 2\cos(\vartheta_0)\sin(\psi_0)T^g{}_{2,3} - 2\cos(\varphi_0)\sin(\psi_0)T^g{}_{3,2} \\ & - 2\cos(\vartheta_0)\cos(\varphi_0)T^g{}_{1,1} \\ & + T^q{}_{1,1}{}^2 - 2\cos(\vartheta_0)\sin(\varphi_0)\sin(\psi_0)T^g{}_{1,2} \\ & - 2\cos(\vartheta_0)\sin(\varphi_0)\cos(\psi_0)T^g{}_{1,3} \\ & - 2\sin(\vartheta_0)\sin(\varphi_0)\sin(\psi_0)T^g{}_{2,2} \\ & - 2\sin(\vartheta_0)\sin(\varphi_0)\cos(\psi_0)T^g{}_{2,3} - 2Y_0 T^g{}_{2,4}\end{aligned}$$

A.3 Structural parameters

$$\begin{aligned}\Delta^j = {} & -2\sin(\theta_i)T^n_{i\ 1,3}T^h_{i-1\ 2,3} - 2\sin(\theta_i)T^n_{i\ 1,2}T^h_{i-1\ 2,2} \\ & - 2\cos(\theta_i)T^n_{i\ 1,4}T^h_{i-1\ 1,4} \\ & - 2\cos(\theta_i)T^n_{i\ 1,1}T^h_{i-1\ 1,1} - 2\cos(\theta_i)T^n_{i\ 1,2}T^h_{i-1\ 1,2} \\ & - 2\sin(\theta_i)T^n_{i\ 1,4}T^h_{i-1\ 2,4} \\ & + d^2 - 2dT^h_{i-1\ 3,4} + T^h_{i-1\ 3,4}{}^2 - 2\cos(\theta_i)T^n_{i\ 1,3}T^h_{i-1\ 1,3} \\ & + T^h_{i-1\ 2,1}{}^2 + T^h_{i-1\ 2,2}{}^2 \\ & + T^h_{i-1\ 3,1}{}^2 + T^h_{i-1\ 1,4}{}^2 + T^h_{i-1\ 1,1}{}^2 + T^h_{i-1\ 1,2}{}^2 + T^h_{i-1\ 1,3}{}^2 \\ & + T^n_{i\ 2,4}{}^2 + T^n_{i\ 3,4}{}^2 T^n_{i\ 1,4}{}^2 \\ & + T^n_{i\ 1,3}{}^2 + a^2 + T^n_{i\ 1,1}{}^2 + T^n_{i\ 2,1}{}^2 + T^n_{i\ 3,1}{}^2 + T^n_{i\ 2,2}{}^2 \\ & + T^n_{i\ 3,2}{}^2 + T^n_{i\ 2,3}{}^2 + T^n_{i\ 3,3}{}^2 \\ & - 2\sin(\theta_i)T^n_{i\ 1,1}T^h_{i-1\ 2,1} + T^h_{i-1\ 3,2}{}^2 + T^h_{i-1\ 3,3}{}^2 + T^h_{i-1\ 2,3}{}^2 \\ & + T^h_{i-1\ 2,4}{}^2 + T^n_{i\ 1,2}{}^2\end{aligned}$$

$$\begin{aligned}DEN^j_\alpha = {} & -2T^n_{i\ 2,4}T^h_{i-1\ 3,4} + 2\cos(\theta_i)T^n_{i\ 3,2}T^h_{i-1\ 2,2} \\ & + 2\cos(\theta_i)T^n_{i\ 3,1}T^h_{i-1\ 2,1} \\ & + 2\cos(\theta_i)T^n_{i\ 3,3}T^h_{i-1\ 2,3} - 2\sin(\theta_i)T^n_{i\ 3,3}T^h_{i-1\ 1,3} \\ & - 2T^n_{i\ 2,1}T^h_{i-1\ 3,1} \\ & - 2T^n_{i\ 2,2}T^h_{i-1\ 3,2} - 2\sin(\theta_i)T^n_{i\ 3,1}T^h_{i-1\ 1,1} - 2T^n_{i\ 2,3}T^h_{i-1\ 3,3} \\ & + 2\cos(\theta_i)T^n_{i\ 3,4}T^h_{i-1\ 2,4} \\ & + 2T^n_{i\ 2,4}d_i - 2\sin(\theta_i)T^n_{i\ 3,2}T^h_{i-1\ 1,2} \\ & - 2\sin(\theta_i)T^n_{i\ 3,4}T^h_{i-1\ 1,4}\end{aligned}$$

$$\begin{aligned}NUM^j_\alpha = {} & -2T^n_{i\ 3,1}T^h_{i-1\ 3,1} - 2T^n_{i\ 3,4}T^h_{i-1\ 3,4} + 2\sin(\theta_i)T^n_{i\ 2,2}T^h_{i-1\ 1,2} \\ & - 2T^n_{i\ 3,2}T^h_{i-1\ 3,2} \\ & + 2T^n_{i\ 3,4}d + 2\sin(\theta_i)T^n_{i\ 2,1}T^h_{i-1\ 1,1} \\ & - 2\cos(\theta_i)T^n_{i\ 2,2}T^h_{i-1\ 2,2} \\ & + 2\sin(\theta_i)T^n_{i\ 2,4}T^h_{i-1\ 1,4} - 2\cos(\theta_i)T^n_{i\ 2,3}T^h_{i-1\ 2,3} \\ & + 2\sin(\theta_i)T^n_{i\ 2,3}T^h_{i-1\ 1,3} \\ & - 2T^n_{i\ 3,3}T^h_{i-1\ 3,3} - 2\cos(\theta_i)T^n_{i\ 2,4}T^h_{i-1\ 2,4} \\ & - 2\cos(\theta_i)T^n_{i\ 2,1}T^h_{i-1\ 2,1}\end{aligned}$$

$$a_m = (2T^n_{i\ 1,4} - 2\cos(\theta_i)T^h_{i-1\ 1,4} - 2\sin(\theta_i)T^h_{i-1\ 2,4})a_i$$

A.4 Joint parameters

$$\begin{aligned}NUM_\theta = {} & -T^n_{i\ 1,1}T^h_{i-1\ 1,1} - T^n_{i\ 1,4}T^h_{i-1\ 1,4} - T^n_{i\ 1,3}T^h_{i-1\ 1,3} \\ & - T^n_{i\ 1,2}T^h_{i-1\ 1,2} \\ & - aT^h_{i-1\ 1,4} + \sin(\alpha_i)T^n_{i\ 3,3}T^h_{i-1\ 2,3} \\ & - \cos(\alpha_i)T^n_{i\ 2,4}T^h_{i-1\ 2,4} \\ & + \sin(\alpha_i)T^n_{i\ 3,4}T^h_{i-1\ 2,4} + \sin(\alpha_i)T^n_{i\ 3,1}T^h_{i-1\ 2,1} \\ & - \cos(\alpha_i)T^n_{i\ 2,2}T^h_{i-1\ 2,2} \\ & - \cos(\alpha_i)T^n_{i\ 2,1}T^h_{i-1\ 2,1} - \cos(\alpha_i)T^n_{i\ 2,3}T^h_{i-1\ 2,3} \\ & + \sin(\alpha_i)T^n_{i\ 3,2}T^h_{i-1\ 2,2}\end{aligned}$$

$$\begin{aligned}DEN_\theta = {} & -a_i T^h_{i-1\ 2,4} - \sin(\alpha_i)T^n_{i\ 3,4}T^h_{i-1\ 1,4} + \cos(\alpha_i)T^n_{i\ 2,4}T^h_{i-1\ 1,4} \\ & - T^n_{i\ 1,3}T^h_{i-1\ 2,3} - T^n_{i\ 1,4}T^h_{i-1\ 2,4} - T^n_{i\ 1,1}T^h_{i-1\ 2,1} \\ & + \cos(\alpha_i)T^n_{i\ 2,3}T^h_{i-1\ 1,3} \\ & - T^n_{i\ 1,2}T^h_{i-1\ 2,2} - \sin(\alpha_i)T^n_{i\ 3,1}T^h_{i-1\ 1,1} \\ & + \cos(\alpha_i)T^n_{i\ 2,2}T^h_{i-1\ 1,2} \\ & + \cos(\alpha_i)T^n_{i\ 2,1}T^h_{i-1\ 1,1} - \sin(\alpha_i)T^n_{i\ 3,2}T^h_{i-1\ 1,2}\end{aligned}$$

$$\begin{aligned}\Delta_\theta = {} & T^n_{i\ 2,1}{}^2 + T^n_{i\ 3,1}{}^2 + T^n_{i\ 2,2}{}^2 + T^n_{i\ 3,2}{}^2 + T^n_{i\ 2,3}{}^2 \\ & + T^n_{i\ 3,3}{}^2 + T^n_{i\ 2,4}{}^2 \\ & + T^n_{i\ 1,3}{}^2 + a^2 + T^n_{i\ 1,4}{}^2 + T^n_{i\ 1,2}{}^2 + T^n_{i\ 1,1}{}^2 + T^n_{i\ 1,4}a_i \\ & - \sin(\alpha_i)T^n_{i\ 3,3}T^h_{i-1\ 1,3}\end{aligned}$$

$$d_m = -\sin(\alpha_i)T^n_{i\ 2,4} - \cos(\alpha_i)T^n_{i\ 3,4} + T^h_{i-1\ 3,4}$$

A.5 Base frame parameters

$$\begin{aligned}NUM_{\vartheta_0} = {} & \sin(\psi_0)T^g{}_{2,2} + \sin(\varphi_0)\sin(\psi_0)T^g{}_{1,2} \\ & + \sin(\varphi_0)\cos(\psi_0)T^g{}_{1,3} - \sin(\psi_0)T^g{}_{2,3} \\ & + \cos(\varphi_0)T^g{}_{1,1}\end{aligned}$$

$$\begin{aligned}DEN_{\vartheta_0} = {} & -\sin(\varphi_0)\sin(\psi_0)T^g{}_{2,2} + \cos(\psi_0)T^g{}_{1,2} \\ & - \sin(\psi_0)T^g{}_{1,3} - \cos(\varphi_0)T^g{}_{2,1} \\ & - \sin(\varphi_0)\cos(\psi_0)T^g{}_{2,3}\end{aligned}$$

$$\begin{aligned}NUM_{\varphi_0} = {} & \sin(\vartheta_0)T^g{}_{2,1} + \cos(\psi_0)T^g{}_{3,3} + \sin(\psi_0)T^g{}_{3,2} \\ & + \cos(\vartheta_0)T^g{}_{1,1} - \sin(\vartheta_0)\cos(\psi_0)T^g{}_{2,3}\end{aligned}$$

$$\begin{aligned}DEN_{\varphi_0} = {} & T^g{}_{3,1} - \cos(\vartheta_0)\sin(\psi_0)T^g{}_{1,2} \\ & - \cos(\vartheta_0)\cos(\psi_0)T^g{}_{1,3} - \sin(\vartheta_0)\sin(\psi_0)T^g{}_{2,2}\end{aligned}$$

$$\begin{aligned}NUM_{\psi_0} = {} & \cos(\varphi_0)T^g{}_{3,3} - \sin(\vartheta_0)T^g{}_{1,2} \\ & + \cos(\vartheta_0)\sin(\varphi_0)T^g{}_{1,3} \\ & + \cos(\vartheta_0)T^g{}_{2,2} + \sin(\vartheta_0)\sin(\varphi_0)T^g{}_{2,3}\end{aligned}$$

$$\begin{aligned}DEN_{\psi_0} = {} & \cos(\vartheta_0)T^g{}_{2,3} - \cos(\varphi_0)T^g{}_{3,2} - \sin(\vartheta_0)T^g{}_{1,3} \\ & - 2\sin(\vartheta_0)\sin(\varphi_0)T^g{}_{2,2} - \cos(\vartheta_0)\sin(\varphi_0)T^g{}_{1,2}\end{aligned}$$

https://doi.org/10.1017/S0263574797000787 Published online by Cambridge University Press

B APPENDIX B

A task unit consists of a task specification and a task body:

```
task_specification::=
    task [type] identifier [is
        {entry_declaration}
        {representation_clause}
    end [task_simple_name]
g
task_body::=
    task body task_simple_name is
        [declarative_part]
    begin
        sequence_of_statements
    [exception
        exception_handler
        {exception_handler}]
    end [task_simple_name]
```

Each agent of the distributed method (a joint) is modeled as a task in this manner:

```
task type AGENT is
    entry INITIALISATION
        NUMBER: in integer;
        PARA: in out parameters
    entry COMPUTE
        MAT_DEST: in MAT_44;
        MAT_OI: in MAT_44
end AGENT
link: array (1..n) of agent;

task body AGENT is
    begin
        accept INITIALISATION (.....) do
            id: NUMBER
        end INITIALISATION;
        loop
            accept COMPUTE (.....)
                if id#1 then
                    MAT_DEST_LOC
                    =INVERSE(MAT_OI)
                    * MAT_DEST (equation...)
                else
                    MAT_DEST_LOC=MAT_DEST
                end if;
                -- Computes PARA
                if id#1 then
                    link (id-1).COMPUTE (.....)
                else
                    link(n).COMPUTE(.....)
                end if;
        end COMPUTE;
        end loop
end body AGENT;
```

The main program SYNTHESIS is:

```
begin
    for i in 1..n loop
        link(i).INITIALISATION (.....)
    end loop;
    link(N).COMPUTE(.....);
end SYNTHESIS
```

https://doi.org/10.1017/S0263574797000787 Published online by Cambridge University Press