A Bayesian CAD system for robotic Applications

Kamel Mekhnacha, Emmanuel Mazer, Pierre Bessiere

A BAYESIAN CAD SYSTEM FOR ROBOTIC APPILCATIONS

KAMEL MEKHNACHA
Leibniz/IMAG
46, Avenue Felix Viallet,
38031 Grenoble Cedex, France
Kamel.Mekhnacha@imag.fr

EMMANUEL MAZER
GRAVIR/IMAG
INRIA Rhne-Alpes, ZIRST,
38330 Montbonnot, France
Emmanuel.Mazer@imag.fr

PIERRE BESSIRE
Leibniz/IMAG
46, Avenue Felix Viallet,
38031 Grenoble Cedex, France
Pierre.Bessiere@imag.fr

ABSTRACT

We present in this paper a Bayesian CAD system for robotic applications. We address the problem of the propagation of geometric uncertainties, and how to take into account this propagation when solving inverse problems. The methodology used to represent and handle uncertainties using conditional probability distributions on on the system's parameters and the sensor measurements is presented. It may be seen as a generalization of constraint-based approaches in which we explicitly model geometric uncertainties. Using this methodology, a constraint is a represented by a probability distribution instead of a simple equality or inequality. Numerical algorithms used to apply this methodology are also described. Using an example, we show how to apply our approach by providing simulation results using our CAD system.

KEYWORDS: Robotics, CAD, Bayesian reasoning, Monte Carlo methods, Geometric constraints.

1 INTRODUCTION

Using geometric models in robotics and CAD systems necessarily requires a more or less realistic modeling of the environment. However, the validity of calculations using these models depends on their degree of fidelity to the real environment and the capacity of these systems to represent and to take into account possible differences between these models and reality when solving a given problem.

This paper presents a methodology based on Bayesian formalism to represent and handle geometric uncertainties in robotics and CAD systems. The approach presented in this paper may be seen as a generalization of constraint-based approaches where uncertainties on models are taken into account. A constraint on a relative pose between two frames is represented by a probability distribution on parameters of this pose instead of a simple equality or inequality.

For a given problem, the marginal distribution on the unknown parameters is inferred using the probability calculus. The original problem is reduced to an optimization problem over the marginal distribution to find a solution with maximum probability. In the general case, this marginal probability may contain an integral on a large dimension space.

The resolution method used for this integration/optimization problem is based on an adaptive genetic algorithm. The problem of integral's estimation is approached using a stochastic Monte Carlo method. The accuracy of this numerical estimation of integrals is controlled by the optimization process to reduce computation time.

Experimental results made on the implemented CAD system have demonstrated the effectiveness and the robustness of our approach. Numerous geometric problems have been specified and resolved using our system, including kinematics inversion under uncertainties, robot calibration, parts' pose and shape calibration, as well as robotic workcell design. An example of this experimentation is presented in this paper.

This paper is organized as follows. We first report related work. In Section 3, we present our specification methodology, and show how to obtain an optimization problem. In Section 4, we describe our numerical resolution method. We present an example to illustrate our approach in Section 5, and give some conclusions and perspectives in Section 6.

2 RELATED WORK

Geometric uncertainties representation and handling is a central issue in the fields of robotics and mechanical assembly. Since the precursor work of Taylor [Taylor, 1976], in which geometric uncertainties were taken into account in the robot manipulators planning process, numerous approaches have been proposed to model these uncertainties explicitly.

Methods modeling the environment using "certainty grids" [Moravec, 1988] and those using uncertain models of motion [Alami and Simeon, 1994] have been extensively used, especially in mobile robotics.

Gaussian models to represent geometric uncertainties and to approximate their propagation have been

proposed in manipulators programming [Puget, 1989] as well as in assembly [Sanderson, 1997]. These methods have the advantage of commodity of the computation they require. However, they are only applicable when a linearization of the model is possible. Another limitation of these methods is their inability to take inequality constraints into account.

Geometric constraint-based approaches [Taylor, 1976, Owen, 1996] using constraint solvers have been used in robotic task-level programming systems. Most of these methods do not represent uncertainties explicitly. They handle uncertainties using a least-squares criterion when the solved constraints systems are over-determined. In the cases where uncertainties are explicitly taken into account (as is the case in Taylor's system), they are solely described as inequality constraints one possible variations.

3 PROBABILISTIC GEOMETRIC CONSTRAINTS SPECIFICATION

In this section, we describe our methodology by giving some concepts and definitions necessary for probabilistic geometric constraints specification. We further show how to obtain an objective function to maximize from the original geometric problem.

3.1 PROBABILISTIC KINEMATIC GRAPH

A geometric problem is described as a "probabilistic kinematic graph", which we define as the directed graph having a set of n frames $S = \{S_1, \cdots, S_n\}$ as vertices and a set of m edges $A = \{A_{i_1 j_1}, \cdots, A_{i_m j_m}\}$, where $A_{i_k j_k}$ denotes an edge between the parent vertex S_{i_k} and its child S_{j_k} and represents a probabilistic constraint on the corresponding relative pose. We call these edges "probabilistic kinematic links". A given edge may describe:

- a modeling constraint (a piece of knowledge) on the relative pose between the parent frame and the child one,
- a sensor measurement on the pose of a given entity,
- or a constraint we wish to satisfy to solve the problem (an objective value with a given precision, for example).

Each edge $A_{i_k j_k}$ is labeled by:

1. a probability distribution $p(Q_{i_k j_k})$ where $Q_{i_k j_k}$ is the relative pose vector (6-vector) $Q_{i_k j_k} = (t_x t_y t_z r_x r_y r_z)^T$. The first three parameters of this 6-vector represent the translation, while the remaining three represent the rotation.
2. possible equality/inequality constraints ($E_k(Q_{i_k j_k}) = 0$, $C_k(Q_{i_k j_k}) \leq 0$). These constraints represent possible geometric relationships between the two geometric entities attached to these two frames. Their shapes depend on the type of the geometric relationship. We implement several relationships between geometric entities in this work, such as points, polygonal faces, edges, spheres and cylinders. The details on equality/inequality constraints induced by these relationships can be found in [Mekhnacha, 1999].
3. a "status" 6-vector describing for each parameter of $Q_{i_k j_k}$, its role (nature) in the problem. A status can take one of the 3 following values:
   - *Unknown* (denoted X) for parameters representing the unknown variables of the problem and whose values must be found to solve the problem.
   - *Free* (denoted L) for parameters whose values are only known with a probability distribution.
   - *Fixed* (denoted F) for parameters having known fixed values that cannot be changed.

In the general case, the kinematic graph may contain a set of cycles. The presence of a cycle represents the existence of more than one path between two vertices (frames) of the graph. To ensure the geometric coherence of the model, the computation of the relative pose between these two frames using all paths must give the same value. For each cycle containing k edges, we must have:

$$\begin{aligned} T_{ii} &= T_{i\ i+1}^{s_i\ i+1} * T_{i+1\ i+2}^{s_{i+1}\ i+2} * \cdots * T_{k-1\ k}^{s_{k-1}\ k} * \\ &\quad T_{k\ 1}^{s_k\ 1} * T_{1\ 2}^{s_1\ 2} * \cdots * T_{i-1\ i}^{s_{i-1}\ i} \\ &= I_4, \end{aligned} \tag{1}$$

where T_{ij} is the 4×4 homogeneous matrix corresponding to the pose vector Q_{ij}, I_4 is the 4×4 identity matrix and $s_{ij} \in \{-1, 1\}$ is the direction in which the edge A_{ij} has been used. We call these additional equality constraints the "cycle-closing constraints". They are global constraints involving, for each cycle, all parameters it contains. The minimal number of cycles allowing coverage of a connected graph having n vertices and m edges is $p = m - n + 1$ (see [Gondran and Minoux, 1990]). Consequently, we obtain p cycle-closing constraints for a given problem.

3.2 OBJECTIVE FUNCTION

Given a probabilistic kinematic graph, we are interested in constructing a marginal distribution over the unknown parameters of the problem. Maximizing this distribution will give a solution to the problem.

To do so, we define the following sets of propositions:

- A set of p propositions $\{\mathcal{K}_i\}_{i=1}^{p}$ such as: $\mathcal{K}_i \equiv$ "cycle c_i is closed".
- A set of m propositions $\{\mathcal{H}_k\}_{k=1}^{m}$ such as: $\mathcal{H}_k \equiv$ "$C_k(Q_{i_k j_k}) \leq 0$ and $E_k(Q_{i_k j_k}) = 0$".

If we denote the unknown parameters of the problem by X, a solution to a problem is a value of X that maximizes the distribution

$$p(X|\mathcal{H}_1 \cdots \mathcal{H}_m \mathcal{K}_1 \cdots \mathcal{K}_p).$$

For each edge A_{ij}, if we denote by L_{ij} the set of parameters having the L status, and by X_{ij} the parameters having the X status, we can write, using the probability calculus and the p cycle-closing constraints (Eq. 1), the following general form:

$$p(X|\mathcal{H}_1 \cdots \mathcal{H}_m \mathcal{K}_1 \cdots \mathcal{K}_p) \propto p(X) I(X),$$

where

$$\begin{aligned} I(X) \quad = \quad & \int dL \\ & p(L_{i_1 j_1}) p(\mathcal{H}_1 | X_{i_1 j_1} L_{i_1 j_1}) \\ & \vdots \\ & p(L_{i_{m-p} j_{m-p}}) p(\mathcal{H}_{m-p} | X_{i_{m-p} j_{m-p}} L_{i_{m-p} j_{m-p}}) \\ & p_{O_1}(F_1(X, L)) p(\mathcal{H}_{m-p+1} | F_1(X, L)) \\ & \vdots \\ & p_{O_p}(F_p(X, L)) p(\mathcal{H}_m | F_p(X, L)). \end{aligned} \tag{2}$$

For each cycle c_i, $i = 1 \cdots p$, O_i denotes a pose vector pertaining to c_i and F_i is the function allowing computation of the value of this pose vector using the values of all other pose vectors pertaining to c_i (using Eq. 1). p_{O_i} denotes the distribution over O_i, while L is the concatenation of $L_{i_1 j_1}, \cdots, L_{i_{m-p} j_{m-p}}$.

4 RESOLUTION METHOD

We described in the previous section how to obtain an integration/optimization problem from a kinematic graph:

$$X^* = \max_X \left[p(X|\mathcal{H}_1 \cdots \mathcal{H}_m \mathcal{K}_1 \cdots \mathcal{K}_p) \right]. \tag{3}$$

In this section, we will present the practical methods we use to approach these problems.

4.1 NUMERICAL INTEGRATION METHOD

in real-world applications where integrands may have complex shapes and integration spaces may have very high dimensionality. Domain subdivision-based methods (such as trapezoidal or Simpson methods) are often used for numerical integration in low-dimensional spaces. However, these techniques are poorly adapted for high-dimensional cases.

Using the Trapeze method for example to estimate a one-dimensional integral using n points, we can demonstrate that the variance of this estimator vary as $\frac{1}{n^2}$. In the case of a d-dimensional space, if we assume that the number of subdivision steps is n for each dimension, the variance will vary as $\frac{1}{n^{2/d}}$.

4.1.1 MONTE CARLO METHODS FOR NUMERICAL ESTIMATION

Monte Carlo methods (MC) are powerful stochastic simulation techniques that may be applied to solve optimization and numerical integration problems in large dimensional spaces. Since their introduction in the physics literature in the 1950s, Monte Carlo methods have been at the origin of the recent Bayesian revolution in applied statistics and related fields, including econometrics (see [Geweke, 1996] for example) and biometrics. Their application in other fields such as image synthesis (see [Keller, 1996]) and mobile robotics (see [Dellaert et al., 1999]) is more recent.

Principles

The principle of using Monte Carlo methods for numerical integration is to approximate the integral

$$I = \int p(x) g(x) \; d^d x, \tag{4}$$

by estimating the expectation of the function $g(x)$ under the distribution $p(x)$

$$I = \int p(x) g(x) \; d^d x = \langle g(x) \rangle. \tag{5}$$

Suppose we are able to get a set of samples $\{x^{(i)}\}_{i=1}^{N}$ (d-vectors) from the distribution $p(x)$, we can use these samples to get the estimator

$$\hat{I} = \frac{1}{N} \sum_i g(x^{(i)}). \tag{6}$$

Clearly, if the vectors $\{x^{(i)}\}_{i=1}^{N}$ are generated from $p(x)$, the variance of the estimator $\hat{I} = \frac{1}{N} \sum_i g(x^{(i)})$ will decrease as $\frac{\sigma^2}{N}$ where σ^2 is the variance of g:

$$\sigma^2 = \int p(x) (g(x) - \hat{g})^2 \; d^d x, \tag{7}$$

and $\hat{g}$ is the expectation of g.

This result is one of the important properties of Monte Carlo methods:

The accuracy of Monte Carlo estimates is independent of the dimensionality of the integration space.

4.1.2 USING MC METHODS FOR OUR APPLICATION

Using an MC method to estimate the integral (2) requires:

1. To sample a set of N points $\{L^{(i)}\}_{i=1}^{N}$ from the prior distribution $p(L)$ such that the sampled points respect local equality/inequality constraints (i.e. $\{\mathcal{H}_i\}_{i=1}^{m-p}$ have the value *true*).

2. To estimate the integral $I(X)$ using the set $\{L^{(i)}\}_{i=1}^{N}$ of points as follows:

$$\hat{I}(X) = \frac{1}{N}\sum_{i=1}^{N} p_{O_1}(F_1(X, L^{(i)}))p(\mathcal{H}_{m-p+1}|F_1(X, L^{(i)})) \vdots p_{O_p}(F_p(X, L^{(i)}))p(\mathcal{H}_m|F_p(X, L^{(i)})).$$

Points sampling
The set of N points used to estimate the integral may be sampled in various ways.

Since parameters pertaining to different kinematic links are independent, we decompose the "state vector" L (concatenation of $L_{i_1 j_1}, \cdots, L_{i_{m-p} j_{m-p}}$) to $m-p$ components $\{L_{i_k j_k}\}_{k=1}^{m-p}$ and apply a local sampling algorithm (see [Geweke, 1996]). Updating the state vector L

$$L^{(t)} = (L^{(t)}_{i_1 j_1}, L^{(t)}_{i_2 j_2}, \cdots, L^{(t)}_{i_k j_k}, \cdots, L^{(t)}_{i_{m-p} j_{m-p}})$$

requires to update only one component $L_{i_k j_k}$

$$L^{(t+1)} = (L^{(t)}_{i_1 j_1}, L^{(t)}_{i_2 j_2}, \cdots, L^{(t+1)}_{i_k j_k}, \cdots, L^{(t)}_{i_{m-p} j_{m-p}}).$$

N iterations of this procedure gve us the set $\{L^{(i)}\}_{i=1}^{N}$ that will be used for estimating the integral.

To update a component $L_{i_k j_k}$ (a set of parameters pertaining to the same pose vector $Q_{i_k j_k}$), we have to take into account possible dependencies between these parameters.

Consequently, to update a component $L_{i_k j_k}$, we have to face two problems.

- **Candidate point sampling**
  A candidate $L^c_{i_k j_k}$ is drawn from the distribution $p(L_{i_k j_k})$. Two cases are possible:
  - We dispose of a direct sampling method from $p(L_{i_k j_k})$. It is the case of uniform and Gaussian distributions for example.
  - We do not dispose of a direct sampling method from $p(L_{i_k j_k})$. Therefore, an indirect sampling method has to be used. In this work, we choose to use a Metropolis sampling algorithm (see [Geweke, 1996]).

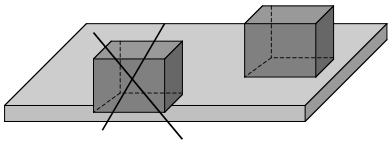

Figure 1: Principle of categorical rejection sampling: the candidate point is rejected because it does not respect the *Face-On-Face* constraint.

- **Candidate validity checking**
  Suppose we have a geometric relationship between to geometric entities E_i and E_j. A geometrical calculus depending on the type of this relationship allows checking the constraint $C_k(Q_{i_k j_k}) \leq 0$. If this constraint is respected (i.e. $p(\mathcal{H}_k|XL_{i_k j_k}) = 1$), the candidate $L^c_{i_k j_k}$ is accepted, otherwise it is rejected .

  Figure 1 shows a *Face-On-Face* relationship example.

4.2 OPTIMIZATION METHOD

The optimization method to be chosen for our application must satisfy a set of criteria in relation to the shape and nature of the function to optimize. The method must:

1. be global, because the function to optimize is often multimodal,

2. allow multiprecision computation of the objective function. Its estimation with high accuracy may require long computation times,

3. allow parallel implementation to improve efficiency.

For our application, we chose a genetic algorithm that satisfies these criteria. First, we present the general principles of these algorithms. Then, we discuss the practical problems we faced when using genetic algorithms in our application, and give the required improvements.

4.2.1 PRINCIPLES OF GENETIC ALGORITHMS

Genetic algorithms (abbreviated GA) are stochastic optimization techniques inspired by the biological evolution of species. Since their introduction by Holland [Holland, 1975] in the seventies, these techniques have been used for numerous global optimization problems, thanks to their ease of implementation and their independence of application fields. They are widely used in a large variety of domains including artificial intelligence [Grefenstette, 1988] and robotics [Mazer et al., 1998].

The goal of a GA is to find a global optimum of a given function F over a search space S.

During an initialization phase, a set of points (*individuals*) are drawn at random from the search space S that is discretized with a given resolution. This set of points is called a *population*.

Each individual I is coded by a string of bits. It represents a solution of the problem and its *adequacy* is measured by a value $F(I)$.

The fundamental principle of genetic algorithms is: "the better the adequacy of an individual, the larger is the probability of selecting it for reproduction". "Genetic operators" are applied to the selected individuals to generate new ones. For a given size of population, better individuals obtained by reproduction replace initial ones. This process is iterated until a convergence criterion is reached.

The standard sequential genetic algorithm can be described as follows. First, an initial population is drawn at random from the search space, and the following cycle is then performed.

1. **Selection:** Using the function F, pairs of individuals are selected. The probability of selecting an individual I grows with the value of $F(I)$ for this individual.

2. **Reproduction:** Genetic operators are applied to the selected individuals to produce new ones.

3. **Evaluation:** The values of F are computed for the new individuals.

4. **Replacement:** Individuals in the current population are replaced by better new individuals.

Many genetic operators are available. However, the more commonly used are "mutation" and "cross-over". For a given pair of individuals, the cross-over operator consists of first cutting the two strings of bits in a randomly chosen place, and then building two new individuals by interchanging the cut parts of the starting strings. The mutation operator consists of flipping some randomly chosen bits of an individual.

In the following, we will use $G(X)$ to denote the objective function $p(X|\mathcal{H}_1 \cdots \mathcal{H}_m \mathcal{K}_1 \cdots \mathcal{K}_p)$.

4.2.2 NARROWNESS OF THE OBJECTIVE FUNCTION – CONSTRAINT RELAXATION

In our applications, the objective function $G(X)$ may have a narrow support (the region where the value is not null) for very constrained problems. The initialization of the population with random individuals from the search space may give null values of the function $G(X)$ for most individuals. This will make the evolution of the algorithm very slow and its behavior will be similar to random exploration.

To deal with this problem, a concept inspired from classical simulated annealing algorithms consists of introducing a notion of "temperature". The principle is to first widen the support of the function by changing the original function to obtain non-null values even for configurations that are not permitted. To do so, we introduce an additional parameter we call T (for temperature) for the objective function $G(X)$. Our goal is to obtain another function $G^T(X)$ that is smoother and has wider support, with

$$\lim_{T \to 0} G^T(X) = G(X).$$

To widen the support of $G(X)$, all elementary terms (distributions) of this later are widened, namely:

- distributions $p_{O_i}(F_i(X, L))$, where $i = 1 \cdots p$.
- inequality constraints $p(\mathcal{H}_j | F_j(X, L))$, where $j = m - p + 1 \cdots m$.

For example:

- for a Gaussian distribution:

$$\begin{aligned} f(x) &= \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}\frac{(x-\mu)^2}{\sigma^2}} \\ f^T(x) &= \frac{1}{\sqrt{2\pi}\sigma(1+T)} e^{-\frac{1}{2}\frac{(x-\mu)^2}{[\sigma(1+T)]^2}} \end{aligned}$$

- for an inequality constraint over the interval $[a, b]$:

$$\begin{aligned} f(x) &= \begin{cases} 1 & \text{if } a \le x \le b \\ 0 & \text{else} \end{cases} \\ f^T(x) &= \begin{cases} 1 & \text{if } a \le x \le b \\ e^{-\frac{(x-a)^2}{(b-a)T}} & \text{if } x < a \\ e^{-\frac{(x-b)^2}{(b-a)T}} & \text{otherwise} \end{cases} \end{aligned}$$

In the general case, inequality constraints may be more complex. Figure 2 shows the case of a *Point-On-Face* inequality constraint for a square face.

4.2.3 ACCURACY OF THE ESTIMATES – MULTI-PRECISION COMPUTING

The second problem we must face is that only an approximation $\hat{G}(X)$ of $G(X)$ is available, of unknown accuracy. Using a large number of points to obtain sufficient accuracy may be very expensive in computation time, so that use of a large number of points in the whole optimization process is inappropriate.

Since the accuracy of the estimate $\hat{G}(X)$ of the objective function depends on the number of points N used for the estimation, we introduce N as an additional parameter to define a new function $\hat{G}_N(X)$.

Suppose we initialize and run for some cycles a genetic algorithm with $\hat{G}_{N_1}(X)$ as evaluation function. The population of this GA is a good initialization for another GA having $\hat{G}_{N_2}(X)$ as evaluation function with $N_2 > N_1$.

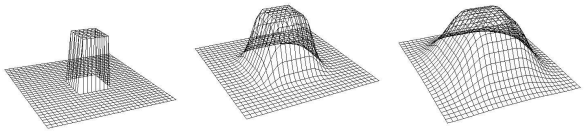

Figure 2: The distribution corresponding to inequality constraints induced by a *Point-On-Face* relationship for a square face at different values of temperature. The left figure shows the original constraints ($T = 0$), while the middle and the right ones show these constraints relaxed at ($T = 50$) and ($T = 100$) respectively.

4.2.4 GENERAL OPTIMIZATION ALGORITHM

In the following, we label the evaluation function (the objective function) by the temperature T and the number N of points used for estimation. It will be denoted by $G_N^T(X)$.

Our optimization algorithm may be described by the following three phases.

1. Initialization and initial temperature determination.
2. Reduction of temperature to recreate the original objective function.
3. Augmentation of the number of points to increase the accuracy of the estimates.

Initialization: The population of the GA is initialized at random from the search space. To minimize computing time in this initialization phase, we use a small number N_0 of points to estimate integrals. We propose the following algorithm as an automatic initialization procedure for the initial temperature T_0, able to adapt to the complexity of the problem.

INITIALIZATION(AG)
BEGIN
FOR each population[i] ∈ AG's population DO
REPEAT
population[i] = random(S)
value[i] = $G_{N_0}^T$(population[i])
if (value[i] == 0.0)
T = T + ΔT
UNTIL (value[i]> 0.0)
FEND
Re-evaluate(population)
END
where ΔT is a small increment value.

Temperature reduction: To obtain the original objective function ($T = 0.0$), a possible scheduling procedure consists of multiplying the temperature, after running the GA for a given number of cycles nc_1, by a factor α ($0 < \alpha < 1$). In this work, the value of α has been experimentally fixed to 0.8. We can summarize the proposed algorithm as follows.

TEMP_REDUCTION(AG)
BEGIN
WHILE (T > T_ϵ) DO
FOR i=1 TO nc_1 DO
Run(AG)
FEND
T = T * α
WEND
T = 0.0
Re-evaluate(population)
END
where T_ϵ is a small threshold value.

Augmenting the number of points: At the end of the temperature reduction phase, the population may contain several possible solutions for the problem. To decide between these solutions, we must increase the accuracy of the estimates. One approach is to multiply N, after running the GA for a given number of cycles nc_2, by a factor β ($\beta > 1$) so that the variance of the estimate is divided by β:

$$Var(G^0_{\beta * N}(X)) = \frac{1}{\beta} Var(G^0_N(X))$$

We can describe this phase by the following algorithm.

N_POINTS_AUGMENTATION(AG)
BEGIN
WHILE (N < N_{max}) DO
FOR i=1 TO nc_2 DO
Run(AG)
FEND
N = N * β
WEND
END
where N_{max} is the number of points that allows convergence of the estimates $\hat{G}^0_N(X)$ for all individuals of the population.

5 EXAMPLE

In this section, we describe how to use our CAD system for concrete problems. We present in detail a calibration problem.

5.1 PROBLEM DESCRIPTION

The purpose of this example is to calibrate the pose and the size of a 3-D part. More precisely, we are interested in identifying the parameters of the pose of

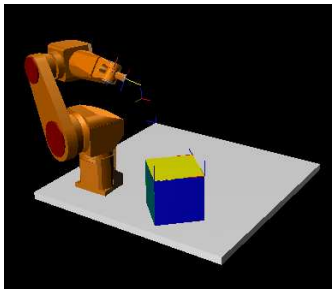

Figure 3: A parallelepiped pose and dimensions calibration problem using contact relationships.

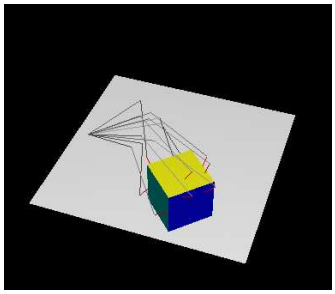

Figure 4: The set of contacts to use for calibration.

a parallelepiped on a table, and the 3 dimensions of this parallelepiped (see Figure 3).

The experimental protocol is as follows. For each measurement, a 6 DOF arm is moved to a configuration that allows obtaining a contact between a touch sensor mounted on on the end effector of the arm, and a face of the parallelepiped. A set of N contacts between the touch sensor and the faces will give the set of N measurements (configurations that allow contact) we will use for calibration (see Figure 4).

The used arm is a 6 DOF Stubli Rx90 manipulator. It is modeled as a set of parts attached to each other using probabilistic links. We especially suppose that we have significant uncertainties on zero positions.

We suppose that the parallelepiped lies on the table. Consequently, we have to identify only the x and y position parameters and the α orientation parameter. For the size of the parallelepiped, we have to identify the parameters sx, sy and sz representing distances between each pair of parallel faces. We used for this example a set of 10 contacts. For each face (except for the inferior face which lies in the table), two measurements have been taken. Figure 5 shows the contact points and the corresponding faces to put back in contact to solve this calibration problem, while Figure 6 gives the kinematic graph corresponding to this problem.

5.2 RESULTS

We summarize the problem complexity and the system performances for this problem using a PowerPC

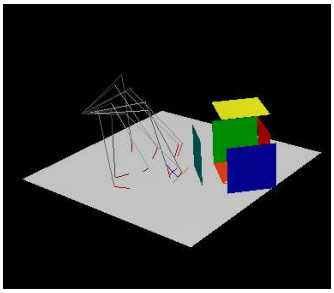

Figure 5: Contact points and parallelepiped faces to put back in contact to solve the calibration problem.

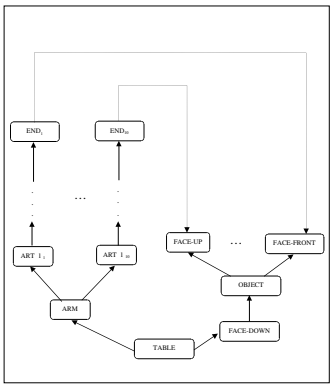

Figure 6: The kinematic graph for the calibration problem.

G3/400 machine in Table 1.

The simulated contacts have been taken at non-null distances between the touch sensor and parallelepiped faces. Table 2 gives error values for the 10 measurements. We have to underline that all these contact errors have positive values because the touch sensor cannot overlap the parallelepiped.

Table 3 gives simulation values of the parameters to calibrate and the values obtained after calibration.

5.3 DISCUSSION

This example presents an application of our method for parameter identification problems. We show especially that using this method allows:

- to take into account prior information on the parameters to estimate.
- to take into account, for each measurement (contact), the accuracy of this later by propagating the uncertainties of the arm model. This allows an implicit weighting of these measurements (the more accurate the measurement, the more importance it has in the calibration process).
- to take into account prior information on the used measurement tool. In this particular example where measurements are contact relation-

	Contact 1	Contact 2	Contact 3	Contact 4	Contact 5
Simulated errors (mm)	0.677	0.567	0.303	0.792	0.724
	Contact 6	Contact 7	Contact 8	Contact 9	Contact 10
Simulated errors (mm)	0.791	0.883	0.858	0.383	0.111

Table 2: Error values used when simulating contacts.

	$x(mm)$	$y(mm)$	$\alpha(rad)$	$sx(mm)$	$sy(mm)$	$sz(mm)$
Simulation values	900.000	-900.000	0.7854	300.000	300.000	300.000
Calibration results	900.195	-900.000	0.7853	299.238	299.238	299.238

Table 3: Initial values (simulation values) of the parameters to calibrate and calibration results.

Integration space dimension	30
Optimization space dimension	6
Number of cycles	10
Number of frames	77
Inequality constraints number	40
Computation time (seconds)	23

Table 1: Some parameters summarizing the problem complexity and the system performances for this calibration problem.

ships, we have expressed the non-overlap phenomenon using a non-symmetrical distribution

$$p(t_z) = \begin{cases} \frac{2}{\sqrt{2\pi}\sigma_c} e^{-\frac{1}{2}\frac{t_z^2}{\sigma_c^2}} & \text{if } t_z \geq 0 \\ 0 & \text{else} \end{cases}$$

where σ_c was $0.5mm$.

6 CONCLUSION

We have presented a generic approach for geometric problem specification and resolution using a Bayesian framework. We have shown how a given problem is first represented as a probabilistic kinematic graph, and then expressed as an integration/optimization problem. Appropriate numerical algorithms used to apply this methodology are also described. For generality, no assumptions have been made on the shapes of distributions or on the amplitudes of uncertainties.

Experimental results made on our system have demonstrated the effectiveness and the robustness of our approach. However, additional studies are required to improve both integration and optimization algorithms.

For the integration problem, numerical integration can be avoided when the integrand is a product of generalized normals (Dirac's delta functions and Gaussians) and when the model is linear or can be linearized (errors are small enough). The optimization algorithm may also be improved by using a local derivative-based method after the convergence of our genetic algorithm. Future work will aim at allowing the use of high-level sensors such as vision-based ones. We are also considering extending our system so that it can include non-geometrical parameters (inertial parameters for example) in problem specification.

References

[Alami and Simeon, 1994] Alami, R. and Simeon, T. (1994). Planning robust motion strategies for mobile robots. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, volume 2, pages 1312–1318, San Diego, California.

[Dellaert et al., 1999] Dellaert, F., Fox, D., Burgard, W., and Thrun, S. (1999). Monte Carlo localization for mobile robots. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, Detroit, MI.

[Geweke, 1996] Geweke, J. (1996). Monte Carlo simulation and numerical integration. In Amman, H., Kendrick, D., and Rust, J., editors, *Handbook of Computational Economics*, volume 13, pages 731–800. Elsevier North-Holland, Amsterdam.

[Gondran and Minoux, 1990] Gondran, M. and Minoux, M. (1990). *Graphes et Algorithmes*. Eyrolle, Paris.

[Grefenstette, 1988] Grefenstette, J. J. (1988). Credit assignment in rule discovery systems based on genetic algorithms. *Machine Learning*, 3:225–245.

[Holland, 1975] Holland, J. H. (1975). *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor, MI.

[Keller, 1996] Keller, A. (1996). The fast calculation of form factors using low discrepancy point sequence. In *Proc. of the 12th Spring Conf. on Computer Graphics*, pages 195–204, Bratislava.

[Mazer et al., 1998] Mazer, E., Ahuactzin, J., and Bessière, P. (1998). The Ariadne's Clew algorithm. *J. Artif. Intellig. Res. (JAIR)*, 9:295–316.

[Mekhnacha, 1999] Mekhnacha, K. (1999). *Méthodes probabilistes Bayesiennes pour la prise en compte des incertitudes géométriques: application à la CAO-robotique*. Thèse de doctorat, Inst. Nat. Polytechnique de Grenoble, Grenoble, France.

[Moravec, 1988] Moravec, H. P. (1988). Sensor fusion in certainty grids for mobile robots. *AI Magazine*, 9(2):61–74.

[Owen, 1996] Owen, J. C. (1996). Constraints on simple geometry in two and three dimensions. *Int. J. of Computational Geometry and Applications*, 6(4):421–434.

[Puget, 1989] Puget, P. (1989). *Vérification-Correction de programme pour la prise en compte des incertitudes en programmation automatique des robots*. Thèse de doctorat, Inst. Nat. Polytechnique de Grenoble, Grenoble, France.

[Sanderson, 1997] Sanderson, A. C. (1997). Assemblability based on maximum likelihood configuration of tolerances. In *Proc. of the IEEE Symposium on Assembly and Task Planning*, Marina del Rey, CA.

[Taylor, 1976] Taylor, R. (1976). *A synthesis of manipulator control programs from task-level specifications*. Ph.d thesis, Stanford University, Computer Science Department.