Modeling Uncertainties In Robot Motions

Aleksandar Timcenko and Peter Allen
Department of Computer Science
Columbia University, New York 10027

Abstract

Dealing with uncertainty is one of the major problems in robotics and one of the main obstacles to populating the world with robots that *do something useful*. This paper offers a new method for modeling uncertainties that exist in a robotic system, based on stochastic differential equations. The benefit of using such a model is that we are then able to capture in a analytic mathematical structure three key points underlying robot motion: 1) the ability to properly express uncertainty within the motion descriptions, 2) the dynamic, changing nature of the task and its constraints, and 3) the idea of establishing a success probability or difficulty index for a task. This paper is an expansion of these ideas, describing the models used and some initial experimental results for two robotic tasks: planning a velocity profile under force and time constraints, and a simple peg-in-hole task. With respect to the dynamic nature of robotic motion tasks, the model of the environment uncertainty that we propose here is "dynamic" rather than "static"; the amount of knowledge about the environment is allowed to change as the robot moves. These results suggest that computational models traditionally found in the "lower" levels in robot systems may have application in the "upper" planning levels as well.

INTRODUCTION

Dealing with uncertainty is one of the major problems in robotics and one of the main obstacles to populating the world with robots that *do something useful*. Some well known motion planning techniques, such as the potential-field method, assume that a robot's sensing, control and knowledge of an environment are perfect. This assumption, albeit never absolutely true, is realistic in non-cluttered environments when the required accuracy in the goal is not critical. The simple — and usually quite sufficient — approach is to slightly "grow" the obstacles and "shrink" the goal in the configuration space to compensate for all present uncertainties. Motions planned under these assumptions are usually called *gross* motions.

Nevertheless, the necessity for a more elaborate treatment of uncertainties exists. Intuitively, by conservatively "growing" the obstacles we may either run out of free space or the goal region may disappear. Thus, we need a planning methodology capable of coping with inherent uncertainties in a more elaborate way. More precisely, we need a tool that allows us to suppress the unwanted effects of different uncertainties — for example, even if our robot "slips" from the prescribed trajectory, we want to be able to guide it towards the goal anyway. Another problem we find is that uncertainties are dynamic; they change over time and position, and we need a mechanism that is capable of expressing and reasoning about time dependent uncertainty. Planning in the presence of uncertainties also poses one additional problem, and that is *recognition* of the goal. Due to sensing inaccuracies, the robot may not be able to recognize that the goal has been attained. The planning system has to make sure that its termination predicate is "strong" enough to prevent getting to the goal without recognizing it.

This paper offers a new method for modeling uncertainties that exist in a robotic system, based on stochastic differential equations. The benefit of using such a model is that we are then able to capture in a analytic mathematical structure three key points underlying robot motion: 1) the ability to properly express uncertainty within the motion descriptions, 2) the dynamic, changing nature of the task and its constraints, and 3) the idea of establishing a success probability or difficulty index for a task. This paper is an expansion of these ideas, describing the models used and some initial experimental results.

We have performed experiments that attempt to quantify the uncertainty in robotic motion control and show how it can be used within our model. The statistical justifiability of the proposed model indicates that it resembles the real nature of the random phenomena that govern the system quite well. More importantly, the method we are about to present offers a way of estimating the variance of different types of uncertainties, thus answering questions about both the qualitative and quantitative nature of uncertainty.

With respect to the dynamic nature of robotic motion tasks, the model of the environment uncertainty that we propose here is "dynamic" rather than "static". That means that the amount of knowledge about the environment is allowed to change as robot moves. If the environment model is built on-line using a robot's sensors, it is natural to assume that the knowledge about the nearby, local neighborhood is more accurate than the knowledge about distant objects. This kind of behavior can be modeled through stochastic differential equations. Since the acquisition of environment models is computationally costly, the increasing variance of a model's uncertainty can be used as a criterion for reexamining the environment and rebuilding its model. This model provides great generality in representing environmental un-

certainties.

With respect to defining a success probability or difficulty index in task planning, the model offers new insights. The notion of a success probability offers one strong candidate for a criterion that a planning algorithm should tend to optimize. The domain of applicability and limitations of the idea of a success probability have yet to be investigated, but we believe they are a fruitful research area. We will present some simple experiments to support this idea.

Certain problems in motion planning are notoriously complex. For example, it is shown in [2] that planning the compliant motion in 3-dimensional configuration space with the presence of uncertainty is in general nondeterministic-exponential time hard. That means that according to the current thinking it would require doubly-exponential time to plan such a motion, which is for all but trivial plans an inconceivable requirement. Nevertheless, one can hope that by "smoothing out" the object and the configuration space, and by applying some wisely chosen analytical techniques, non-trivial plans can be made before the end of time. Thus, computational models traditionally found in the "lower" levels in robot systems may have application in "upper" planning levels as well. We may ask if it is possible to use the *predictive* strength of analytical models instead of traditional search techniques. Is it possible to exploit the smooth, differentiable topological structure of configuration space and populate it with mathematical entities that lead to plans as solutions of certain differential equations? These are the questions we want to address in our future work, and this paper offers some evidence that they may have positive answers.

OVERVIEW OF UNCERTAINTY MODELS

Significant work in robotic planning in the presence of uncertainties has been done by Lozano-Pérez and colleagues [11, 5, 3]. It recognizes three main sources of uncertainties present in robotic tasks [3]:

- *sensor uncertainty*, caused by imperfection of the sensory equipment
- *control uncertainty*, caused by an imperfection of the control system
- *environment uncertainty*, caused by the inaccuracy of the world description at the system's disposal

These three types of uncertainties are also recognized by researchers in the area of sensor fusion [9, 14]. Sensor fusion represents a set of methodologies for information retrieval, combination, verification and decision-making. An intelligent sensor system should be capable of autonomously analyzing the situation, estimating the cost of gathering more information versus the quality of already acquired information, and making an appropriate decision. Furthermore, a sensor system has to be able to cope with uncertainties present in raw sensor measurements, as well as with the "holes" in its knowledge about the environment. The sources of errors are usually categorized into the following groups (based on [9]):

- *statistical errors*, due to the random noise in the measurement process.
- *calibration errors*, due to the inaccurate values of the system's static parameters.
- *model errors*, due to the insufficient or inaccurate environment models

These sources of errors can be related to the previously listed uncertainties that planning algorithms have to face. The source of the control uncertainty is mainly the calibration inaccuracy. The sensor uncertainty is modeled as a random process that has a statistical nature.

This brief comparison between the recognized sources of errors in motion planning and sensor fusion leads to the conclusion that there are basically three different types of uncertainties in a robotic system that affect all aspects of a planning and control process. We will, as it is common in motion planning literature, call them sensor, control and environment uncertainties. In the following few paragraphs we will review some of the the "classical" uncertainty models.

Sensor Uncertainty

Sensor uncertainty is caused by the imperfection of the sensory system. There is a numerous literature that addresses this problem. The first question in "sensory integration" is, according to [9], to identify what is being observed and how accurate those observations are.

The model of sensor uncertainty, as given in [12], is the ball $S(\mathbf{q}_0^0, \varepsilon_{\mathbf{q}_0^0})$ in the configuration space $\mathcal{C}$, centered in the actual position $\mathbf{q}_0^0$ and with the radius $\varepsilon_{\mathbf{q}_0^0}$. It defines the set of possible and uniformly distributed measurements of a robot's position $\mathbf{q}_0^c$ by its sensors. Mathematically, this can be expressed as $\mathbf{q}_0^c \in S(\mathbf{q}_0^0, \varepsilon_{\mathbf{q}_0^0})$. In the language of probability theory, $\mathbf{q}_0^c$ is the random variable whose probability distribution is uniform, centered in $\mathbf{q}_0^0$ and with the radius $\varepsilon_{\mathbf{q}_0^0}$ — that is, $\mathbf{q}_0^c \sim \mathcal{U}(\mathbf{q}_0^0, \varepsilon_{\mathbf{q}_0^0})$ where $\mathcal{U}$ denotes the uniform distribution. The probability distribution density function $\psi_{\mathbf{q}_0^c}$ of $\mathbf{q}_0^c$ can be expressed as

$$\psi_{\mathbf{q}_0^c}(\mathbf{q}) = \begin{cases} \frac{1}{\text{volume of } S(\mathbf{q}_0^0, \varepsilon_{\mathbf{q}_0^0})}, & \|\mathbf{q} - \mathbf{q}_0^0\| \le \varepsilon_{\mathbf{q}_0^0} \\ 0, & \|\mathbf{q} - \mathbf{q}_0^0\| > \varepsilon_{\mathbf{q}_0^0} \end{cases}$$

In sensor system-oriented robotics literature more elaborate models of sensor uncertainties can be found. The generalization of the aforementioned model that we will adopt henceforth will entail an arbitrary probability density function $\psi_{\mathbf{q}_0^c}$.

Control Uncertainty

The effects of control uncertainty will be analyzed separately from the effects of sensor uncertainty. That means that in this paragraph we will assume that *there is no sensor uncertainty.*

The usual model of the control uncertainty is the "uncertainty cone" [12, 5]. It is assumed that the *effective* commanded velocity $\mathbf{v}^m$ lies inside the ball with the radius $\varepsilon_{\mathbf{v}^c}$ centered in the *desired* commanded velocity $\mathbf{v}^c$. Since the position $\mathbf{q}_t^m$ in the configuration space $\mathcal{C}$ is given as an integral of the velocity, $\mathbf{q}_t^m = \int \mathbf{v}^m dt$, it turns out that the effective positions conveyed to the robot controller lie inside the *velocity cone*, denoted $B(\mathbf{q}_0^c, \mathbf{v}^c, \varepsilon_{\mathbf{q}_0^c})$ (see figure 1). Notation $B(\mathbf{q}_0^c, \mathbf{v}^c, \varepsilon_{\mathbf{q}_0^c})$ stands for a cone with an apex in $\mathbf{q}_0^c$, a principal axis in direction $\mathbf{v}^c$ and a central angle in the apex of $2 \arcsin \varepsilon_{\mathbf{q}_0^c}$. The apex of the cone is placed in the initial position $\mathbf{q}_0^c = \mathbf{q}_0^m$.

The important underlying assumption in the "velocity cone" model is that the probability distribution inside the

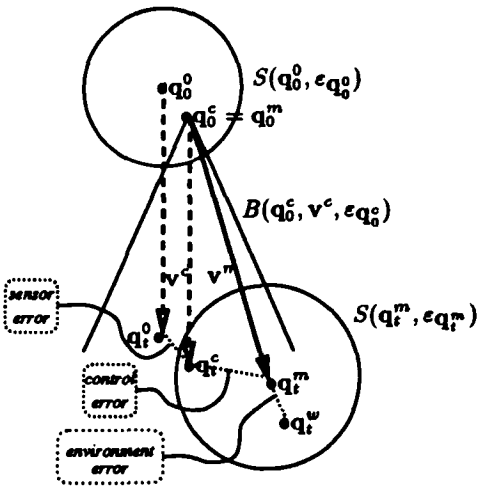

Figure 1: The uncertainty model based on uniform distribution

- q_0^0, q_t^0 are the *nominal* positions in time instances 0 and t (under the assumption that there are no uncertainties in the system)
- q_0^c, q_t^c are the *positions that would be retrieved by a sensor system* in time instances 0 and t under the *desired* commanded velocity (i.e. under the assumption that the only uncertainty present in the system is the sensor uncertainty)
- $S(q_0^0, \epsilon_{q_0^0})$ is the sensor uncertainty sphere
- v^c is the *desired* commanded velocity
- v^m is the *effective* commanded velocity, the velocity actually conveyed to the robot controller
- q_0^m, q_t^m are the *effective* positions in time instances 0 and t (robot positions as a result of sensor and control uncertainties combined)
- $B(q_0^c, v^c, \epsilon_{q_0^c})$ is the *velocity cone*
- q_t^w is the *actual* position (the combination of all three uncertainties)
- $S(q_t^m, \epsilon_{q_t^m})$ is the *environment uncertainty sphere*
- *sensor uncertainty* is given by the displacement $\overline{q_t^0 q_t^c}$
- *control uncertainty* is given by the displacement $\overline{q_t^c q_t^m}$
- *environment uncertainty* is given by the displacement $\overline{q_t^m q_t^w}$

cone is uniform, meaning that all directions inside the cone are equally probable, and that directions outside the cone are impossible. This is an approximation which has its foundations in its simplicity and efficiency in modeling the uncertainty.

Environment Uncertainty

Although the assumption that the planner possesses the complete knowledge about the environment is for all but the most simple tasks unrealistic, the modeling of uncertainties present in the environment description that is at the system's disposal has received relatively small attention. That fact is probably due to the intrinsic difficulties in introducing randomness in geometrical descriptions of the environment. Although sometimes used, terms "uncertain geometry" or "probabilistic geometry" are not adequate notation for the set of tools that are needed for these purposes, mainly because they refer to branches of mathematics that are inherently ill-defined (cf. Bertrand's paradox[1]). Nevertheless, there have been some attempts to theoretically address model uncertainties. A noteworthy work is [4]. Here, an attempt was made to introduce an uncertainty of a geometric object through the distribution function $\psi(\mathbf{p})$ where $\mathbf{p}$ is the parameter vector that describes the object's features. The hidden trap in this approach — analogously to the Bertrand's paradox — is that the statistical nature of the object's features as random variables *depends on the parameterization* $\mathbf{p}$ of the object. It has been recognized in this work that the uniform distribution as a model of uncertainties has its limitations.

An interesting idea of "many universes" is exploited in the model uncertainty description presented in [3]. The configuration space $\mathcal{C}$ is observed as a "slice" (i.e. a subspace) of a broader *generalized configuration space* $\mathcal{C} \times \mathcal{P}_{\mathrm{env}}$ where $\mathcal{P}_{\mathrm{env}}$ is, in words from [3], "an arbitrary index set which parameterizes the model error". At any given time instant the robot is in one of the "slices", although it doesn't know in which. The motion across "slices" is prohibited, since that would mean changing the geometry of the environment. In figure 1 we have depicted the model error by a sphere $S(q_t^m, \epsilon_{q_t^m})$. This means that robot's real position q^w belongs to that sphere. The generalized configuration space is a product space of the uncertainty sphere and the (original) configuration space: $\mathcal{C} \times S(q_t^m, \epsilon_{q_t^m})$. The radius $\epsilon_{q_t^m}$ of the uncertainty sphere depends on q_t^m and varies as the knowledge about the environment varies.

AN UNCERTAINTY MODEL BASED ON STOCHASTIC DIFFERENTIAL EQUATIONS

The guiding idea in this work was to find a unifying model of all three types of uncertainties that is expressive enough to accommodate for most observed phenomena, yet manageable so that it can be used as a basis for motion planning. We propose a model based on stochastic differential equations, developed in the remaining part of this section. This model is a generalization of the "classical" uncertainty

[1]The probability that a chord randomly drawn in a circle is longer than circle's radius depends on the way we define random drawing. This ambiguity is called Bertrand's paradox.

model (which is based on uniformly distributed random variables).

In the next few paragraphs, we will adopt the model for sensor uncertainty, explain the experiment that has been conducted in order to retrieve the nature of the control uncertainty, model that uncertainty by a stochastic differential equation, verify the model and estimate its parameters through a statistical test, and present the environment model of the same type.

Sensor Uncertainty

For the purposes of this paper, we will assume sensor uncertainty is modeled by a known distribution function $\psi_{\mathbf{q}_i^c}$. Due to its simplicity, the natural choice for ψ is Gaussian distribution:

$$\psi_{\mathbf{q}_i^c}(\mathbf{q}) = \frac{1}{(2\pi)^{n/2}(\det\Sigma)^{1/2}} e^{-(\mathbf{q}-\mathbf{q}_i^0)^T\Sigma^{-1}(\mathbf{q}-\mathbf{q}_i^0)/2}$$

where n is the dimensionality of the configuration space (i.e. the dimensionality of $\mathbf{q}$) and Σ is the covariance matrix. Characterizing the actual sensor error is a difficult and important problem that is the subject of ongoing research (see [14, 9, 10])

Control Uncertainty

Before we develop the model of the control uncertainty, we will present an experiment that was used to analyze its nature. It will turn out that the measured data comply to the theoretical model in a statistical test that we have conducted. That implies that our model accurately describes the random phenomenon of control uncertainty.

Experimental Analysis of Control Uncertainty

The experimental setup for investigating the nature of the control uncertainty was as follows. A Sun workstation pointing device ("mouse") was placed in the gripper of a PUMA-560 and positioned directly above the mouse pad. The dimensions of the mouse pad were approximately 6 by 8 inches. Straight-line motion in the xy plane was commanded in 16 different directions, with angular differences of $\pi/8$ radians. The length of each motion was approximately 5 inches. Figure 2 shows one example run. Black lines represent actually observed motion of the pointing device, while gray lines are ideal desired trajectories. Concentric circles are drawn for reference. The experiment has been conducted several times in three different positions: close to the inner boundary of the work space, in the middle of the work space and close to the outer boundary of the work space. The displacements from the ideal trajectory are registered for each commanded direction for different trajectory lengths. The histogram of the displacements in all directions for the 5 inch trajectory length are given in figure 3. This figure indicates that the nature of the random displacements is Gaussian rather than uniform. Secondly, we have experimentally observed that the variances of the displacements increases with the trajectory length. This observation, combined with the similar observations for other trajectory lengths, leads us to make the following two hypotheses:

- the control uncertainty is modeled by a normal distribution
- the variance of the displacements introduced by the control uncertainty rises with the trajectory length

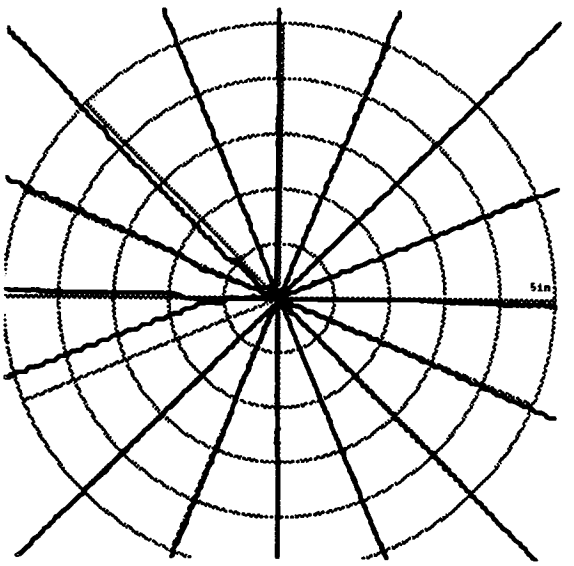

Figure 2: Motions in different directions. Black lines represent observed positions of the pointing device. Gray lines are desired trajectories. Concentric circles are drawn with 1 inch increments in radius. The displacements from the ideal (desired) trajectories are measured along those circles.

From the modeling perspective, there are several reasons for these assumptions. Firstly, the Gaussian distribution is a solution of the linear stochastic differential equation with constant coefficients. In that sense, that is the simplest possible case. Secondly, the changing variance assumption is, as stated in the introduction, a phenomenon that exists in both control and environment uncertainties. Rephrased, the two assumptions from above may read as follows: our model should be as simple as possible (i.e. linear with constant coefficients) and should model the phenomena we have observed (i.e. the increasing variance). The next sections formulate the model and measure how well it agrees with some robotic motion tasks.

The Model Let the control uncertainty be modeled by a stochastic differential equation

$$d\mathbf{q}^m = d\mathbf{q}^c + \sigma^m dW^m \quad (1)$$

Let us try to justify this model. We have assumed earlier that the velocity $\mathbf{v}^m$ lies inside the sphere centered in $\mathbf{v}^c$. Now we will reformulate that assumption: let $\mathbf{v}^m$ be a random variable obtained by superimposing additional noise on $\mathbf{v}^c$:

$$\mathbf{v}^m = \mathbf{v}^c + W \quad (2)$$

where W is the noise component (a Wiener random process). Since $\mathbf{v}^m = \dot{\mathbf{q}}^m$ and $\mathbf{v}^c = \dot{\mathbf{q}}^c$ (dot denotes time differenti-

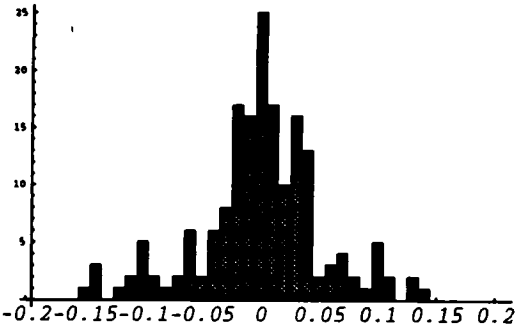

Figure 3: Histogram of the radial displacements of the observed points from the ideal points in all 16 directions, over all experimental runs. The trajectory was 5 inches long. Horizontal axis represents the displacement in inches, and vertical the cumulative number of points in 0.005 inch wide buckets. The total number of points is 240 (15 runs, each contributing 16 points).

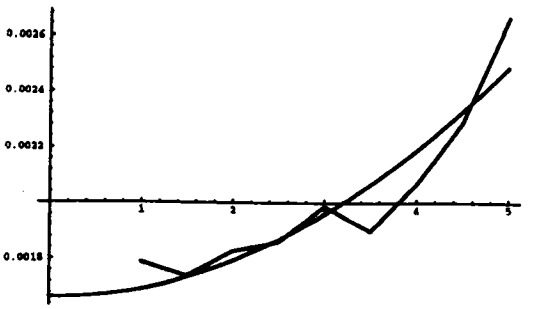

Figure 4: The variances of measured data versus the least-square fit of the parabola. The horizontal axis denotes the distance traveled in inches and the vertical axis is the variance in inches squared.

ation) after multiplying left and right side of 2 by dt, it becomes

$$d\mathbf{q}^m = d\mathbf{q}^c + \sigma^m dW^m$$

where W^m is another Wiener process (appropriately scaled so that it has correct dimensionality) and σ^m a constant matrix that determines the amount of noise in the mapping from $\mathbf{q}^c$ into $\mathbf{q}^m$. σ^m's dimension is the square root of the length. If $\sigma^m = 0$ that mapping would be completely deterministic. Since we have assumed that $\mathbf{q}_0^m = \mathbf{q}_0^c$ we would have that $\mathbf{q}_t^m = \mathbf{q}_t^c$ for any time instant t. However, if $\sigma^m \neq 0$, the mapping from $\mathbf{q}^c$ into $\mathbf{q}^m$ is nondeterministic.

In general, σ^m is not necessarily a constant. We have assumed that it is, and that is for several reasons. Firstly, it simplifies the model and allows for an analytical solution. The assumption that σ^m is constant is equivalent to having a constant radii of the uncertainty spheres. Most of all, we will show shortly that that this assumption models the measured data accurately enough.

The type of solution of equation 1 we are interested in is a probability density function $\psi_{\mathbf{q}_t^m}$ of the random variable $\mathbf{q}_t^m$. In appendix A we derive formula for $\psi_{\mathbf{q}_t^m}$ in the scalar case:

$$\psi_{q_t^m}(q) = \frac{1}{\sqrt{2\pi}\sigma^m\sqrt{q_t^c - q_0^c}} e^{-\frac{(q-q_t^c)^2}{2\sigma^{m\,2}(q_t^c - q_0^c)}} \qquad (3)$$

Thus, q_t^m is normally distributed, $q_t^m \sim \mathcal{N}\left(q_t^c, \sigma^{m\,2}(q_t^c - q_0^c)\right)$, with expectation q_t^c and variance $\sigma^{m\,2}(q_t^c - q_0^c)$. This means that as the robot moves further from the initial point, the uncertainty of its position increases. With the assumption that σ^m is diagonal, the vector case is easily derived. The example for a 2D configuration space is given in next section.

The model derived in the previous paragraph was tested against the experimentally obtained data. We have assumed that the discretization error introduces additional Gaussian noise with variance $\sigma^{c\,2}$. Knowing that the summation of two Gaussian random variables results in another Gaussian variable with a variance equal to the sum of the variances of addends, by combining $\sigma^{c\,2}$ with the relation 3 we obtain the theoretical model for the variance σ_t^2 of the measured data:

$$\sigma_t^2 = \sigma^{c\,2} + \sigma^{m\,2}(q_t^c - q_0^c) \qquad (4)$$

Figure 4 shows the measured variances (computed by the formula $\mathbf{E}(q_t^{m\,2}) - (\mathbf{E}q_t^m)^2$) versus the least-square fit of the parabola of the form 4. Figure 5 shows the χ^2 test of the hypothesis that the data are modeled by normal distributions with zero mean and variance given by 4. The statistics are significant in two cases (2in and 5in) and insignificant in all other cases with a confidence level of 0.9. The largest difference between estimated and measured data is for 3.5 inch trajectory. It is probably caused by spurious data that was way off the expected position. It may indicate that the number of points gathered (240) was not large enough. From this experiment, we can conclude that our model reflects the apparent control uncertainty of this task.

One last comment on the discretization error induced by a mouse pad. The size of the area that covers 900×1152 grid of pixels on the screen is about 6 by 8 inches. That means that the pad's pixel size is approximately 0.007×0.007 inches, so that the average discretization error is about 0.0035 inches.

Environment Uncertainty The environment uncertainty can also be modeled in a similar way to the control uncertainty, which forms part of our overall unifying uncertainty structure. The major difference between the control and environment uncertainty is that the environment uncertainty is a function of the robot's current position. This means that the variance of the model uncertainty varies as the knowledge about the environment varies. This can cause some problems in solving the equations, but there are theoretical methods available to solve for the functional relation between position and variance. [2]

[2]The difficulty that fact introduces is that adjoined backward Kolmogorov equation cannot be solved in general in closed form using traditional methods. However, the solution can be expressed by functional ("Feynman") integrals, using Kac's formula [8].

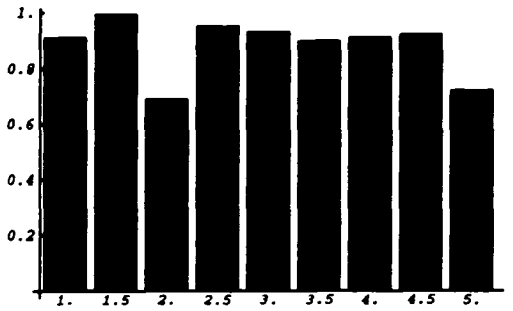

Figure 5: The confidence levels for distribution fits for all observed distances. Horizontal axis is the distance traveled in inches and the vertical axes is the confidence level obtained by a χ^2 test.

The environment uncertainty, in accordance to relation 1, will be modeled by a stochastic differential equation

$$dq^w = dq^m + \sigma^w(q_t^0)dW^w \quad (5)$$

where W^w is a Wiener process and $\sigma^w(q_t^0)$ is the function of the nominal position q_t^0 that describes the amount of model "noise" in any given point.

In the experiments we have conducted we have assumed that the variance $\sigma^w(q_t^0)$ is constant with position, since we do not have software tools to solve for a non-constant case yet.

Putting It Together Putting together all three components of the uncertainty model, we obtain the following stochastic system:

$$\begin{aligned} dq_t^w &= dq_t^m + \sigma^w(q_t^0)dW^w \\ dq_t^m &= dq_t^c + \sigma^m dW^m \\ dq_t^c &= dq_t^0 = v^0 dt \end{aligned}$$

where v^0 is the desired velocity. Note that $v^0 = v^c$. The initial conditions are:

$$\begin{aligned} q_0^c = q_0^m &\sim \mathcal{N}(q_0^0, \sigma^{c\,2}) \\ q_0^w &\sim \mathcal{N}(q_0^0, \sigma_{q_0^0}^{w\,2}) \end{aligned}$$

Thus, the overall uncertainty model is defined by three constants $(\sigma^m, \sigma^c, \sigma_{q_0^0}^w)$ and one function that describes the environment uncertainty (σ^w). A point in the configuration space is thus represented by a random vector with Gaussian distribution. We have assumed that all directions are independent and have the same variance. We will call this model "the continuous uncertainty model".

PLANNING OF A VELOCITY PROFILE UNDER UNCERTAINTY

In this section, we will describe the application of our method to planning velocity profiles for constrained motion amidst obstacles. In particular, this method allows us to compute a success probability that we can use as an optimization criteria for planning a velocity profile in a cluttered and uncertain environment.

Let us consider a simple task of moving along the prescribed path until colliding with an obstacle, and then exerting a prescribed force on the surface of the obstacle. It mainly consists of three phases: moving along the path, colliding, and maintaining the prescribed force. The problem encountered in practice is a manipulator's tendency to bounce from the surface upon initial collision, especially in the case of a very rigid obstacle. That phenomenon is sometimes referred to as a "dynamical instability" and it is shown that simple spring control cannot successfully cope with that problem. Essentially, it is caused by the necessity to *instantly* change the characteristics of motion; in our example to stop and exert a force. Since the manipulator system is not capable of stopping instantly after collision due to its internal delays, it bounces and approaches the obstacle again. If the spring constant is high enough it will bounce again and keep doing that forever. That instantaneousness is the core of the problem: something has to be rapidly changed, and the system might not be able to perform that.

The system's knowledge about the environment is based on models provided by a programmer, and those models are obtained by quantitatively describing the positions and dimensions, as well as other characteristics of objects which constitute the environment. The more accurate those models are, the system can — at least theoretically — utilize that knowledge more efficiently in order to attain the goal of the task more accurately. Let us go back to our force control example for an illustration. If the knowledge of the environment is exact, that is, if the position and the elasticity coefficient of the obstacle are known, the system could move with the maximum speed to the point of contact, computed such that it inflicts the elastic deformation of the obstacle proportional to the required force. On the other extreme, if the knowledge about the environment is zero, the system has to slowly wander through the darkness until it encounters the obstacle, and then to utilize a certain control scheme for maintaining a given force, based on force measurements.

The reality is somewhere in between. The knowledge which is at the system's disposal may be substantial, yet not enough to guarantee that the "full knowledge" strategy is a reasonable choice. We may assume that it is quite unlikely that the obstacle is in a certain region, thus allowing the robot to pass through that region swiftly, while slowing down in the region where the obstacle is expected to be. That means that parameters we can control (velocity in this example) depend on the overall uncertainty of the system and the environment. So, given a velocity, we can compute the probability that the system will fulfill a task within a pre-determined set of constraints such as maximum time for the total motion and maximal impact force upon contact. Our method is to find a velocity at each step of the motion that maximizes the success probability (defined below) and link these into a overall velocity profile for the task given the constraints.

The experiment that we have conducted to demonstrate the use of a success probability in velocity profile planning consists of moving until reaching an obstacle, and exerting a given force after the impact. Let us impose two requirements on our system: the total elapsed time of motion before the impact should be at most τ_{max}, and the maximal force exerted upon contact should not exceed f_{max}. Those two requirements are contradictory: while the former requires the velocity to be high, the latter pushes it back.

Let us define the following binary events (these will allow

us to cast this task as a compound binary predicate):

S = success

TT = obstacle reached in time $\leq \tau_{max}$

F = impact force $\leq f_{max}$

IM = impact has occurred

We will define the probability of success, $\Psi\{S\}$, as an intersection of two events: getting to the goal in time, and not exceeding the maximal force:

$$\Psi\{S\} = \Psi\{TT \cap F\}$$

Applying simple set algebra we have the following relations:

$$\Psi\{F\} = \Psi\{F \cap IM \cup F \cap \overline{IM}\}$$

$$(F \cap IM) \cap (F \cap \overline{IM}) = \emptyset \Rightarrow$$

$$\Psi\{F \cap IM \cup F \cap \overline{IM}\} = \Psi\{F \cap IM\} + \Psi\{F \cap \overline{IM}\}$$

$$\Psi\{F \cap \overline{IM}\} = \Psi\{F|\overline{IM}\}\Psi\{\overline{IM}\}$$

Since $\Psi\{F|\overline{IM}\} = 1$ (the probability of not exceeding the force under the assumption that the impact has not occurred is 1 - the obstacle is simply not reached yet):

$$\Psi\{F \cap \overline{IM}\} = \Psi\{\overline{IM}\}$$

$$\Psi\{F\} = \Psi\{F \cap IM\} + \Psi\{\overline{IM}\} =$$

$$= \Psi\{F|IM\}\Psi\{IM\} + 1 - \Psi\{IM\}$$

$$\Psi\{F\} = 1 - \Psi\{IM\}(1 - \Psi\{F|IM\})$$

Thus we have written the success probability $\Psi\{S\}$ as a function of three probabilities: $\Psi\{TT\}$, $\Psi\{IM\}$ and $\Psi\{F|IM\}$. Using the uncertainty model as described in section 2, one can calculate these probabilities and in each planning step find the velocity that maximizes the success probability. Without going into implementation details, we present the results here.

In figure 6 using our model, we have planned a trajectory that has optimized the success probability for the impact task. The darker areas of the figure are areas of low success probability, and the generated path avoids these areas. Figure 7 is the actual data recorded from a PUMA-560 with wrist sensor that was given a certain motion duration ($\leq \tau_{max}$), and impact force to be minimized ($\leq f_{max}$), in the presence of the environment uncertainty. The velocity profile that maximizes the success probability has the shape that one would intuitively expect: in the area where the obstacle is unlikely to be, the robot starts with a high negative velocity (negative velocity since the direction of movement is downward) and then slows down in order to have a controlled impact force upon the collision. Thus, we can precompute velocity profiles using our model that are able to be mapped into actual robot control strategies.

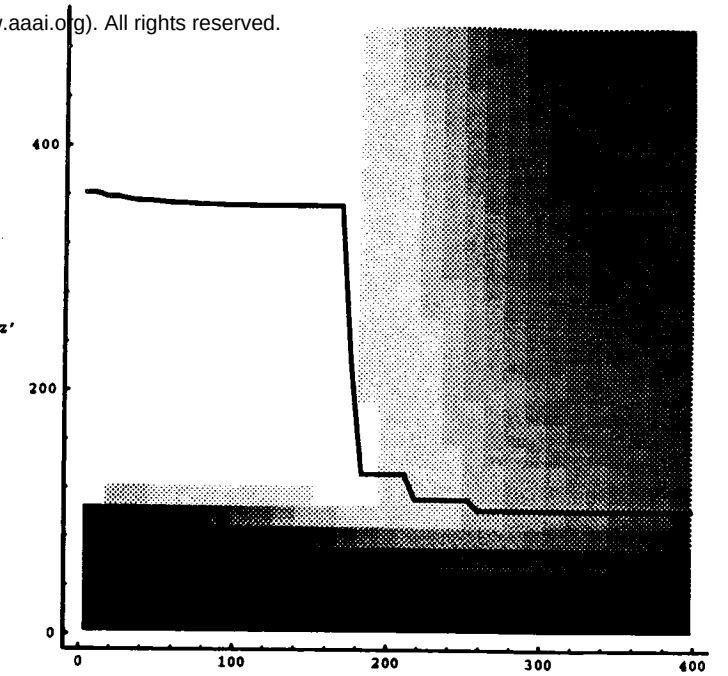

Figure 6: Modeled trajectory using the uncertainty model. Darker areas mean a low probability of success, lighter mean a higher probability of success. For example, low velocities will not allow task completion in the required elapsed time($\leq \tau_{max}$). High velocities may cause impact greater than f_{max}.

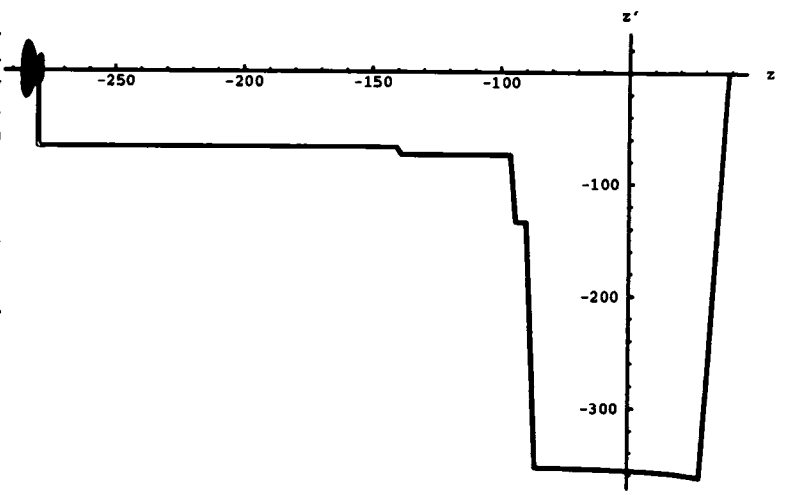

Figure 7: The velocity profile planned so that the motion duration and the impact force are minimized at the same time. The horizontal axis is the position and the vertical axis is the velocity. Note that the motion is "downwards", thus we have negative velocity.

PLANNING A PEG-IN-HOLE TASK UNDER UNCERTAINTY

In this section we will investigate the applicability of the uncertainty model derived earlier to a simple peg-in-hole planning task. The planning problem we consider is the following (see figure 8). Let $\mathcal{C}$ be two-dimensional configuration space that consists of a free space $\mathcal{C}_f$ and a polygonal obstacle $\mathcal{CB}$. As we will see, the requirement that the configuration space is two-dimensional is not essential for the planning algorithm that we will present and stems mainly

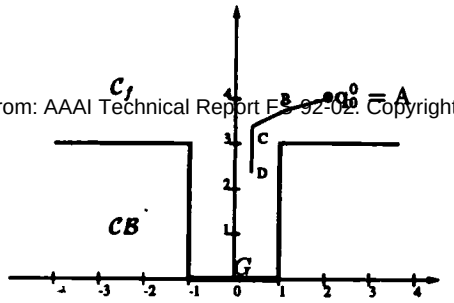

Figure 8: The task is to get to the goal region G starting from $\mathbf{q}_0^0$.

from the ease of visualization. There are two types of motion allowed: "free-flying" motion through the interior of $\mathcal{C}_f$ and the compliant motion along the obstacle's boundary $\partial\mathcal{CB}$. The usual way to model the compliant motion is to assume that robot's controller behaves as a *generalized damper* [15, 13]. The generalized damper guarantees that upon contact the motion resumes in directions orthogonal to the direction of a reactive force. The friction force is modeled by a "friction cone" [12, 5] and there are two possible outcomes upon the impact with an obstacle: the robot either "sticks" and stops (if the velocity points inside the velocity cone) or slides along the obstacle. The task is as follows: starting from the initial configuration $\mathbf{q}_0^0 \in \mathcal{C}_f$, plan the trajectory so that it ends by sticking on a given edge (goal edge, denoted G) from the obstacle's boundary $\partial\mathcal{CB}$ (see figure 8). The known parameters that model the environment are: the friction coefficient μ, the sensor uncertainty σ^c and the control uncertainty σ^m. For now, we will assume that the environment uncertainty does not exist(we hope to address this issue in future research).

The robot's position at time instant t is modeled by a random variable $\mathbf{q}_t^m$ that has a Gaussian distribution with the mean $\mathbf{q}_t^0$ and the variance σ_t^2:

$$\mathbf{q}_t^m \sim \mathcal{N}(\mathbf{q}_t^0, \sigma_t^2)$$

Thus, the distribution density function $\psi_{\mathbf{q}_t^m}(\mathbf{q})$ is given by

$$\psi_{\mathbf{q}_t^m}(\mathbf{q}) = \frac{1}{2\pi\sigma_t^2} e^{-\frac{\|\mathbf{q}_t^0 - \mathbf{q}\|^2}{2\sigma_t^2}} \qquad (6)$$

Let the commanded velocity at time instant t be $\mathbf{v}_t^c$. Due to the control uncertainty, the robot's position after some short period of time Δt will be a random variable $\mathbf{q}_{t+\Delta t}^m$ whose distribution is:

$$\mathbf{q}_{t+\Delta t}^m \sim \mathcal{N}(\mathbf{q}_t^0 + \mathbf{v}_t^c \Delta t, \sigma_t^2 + \sigma^m \|\mathbf{v}_t^c\| \Delta t)$$

The related density is

$$\psi_{\mathbf{q}_{t+\Delta t}^m}(\mathbf{q}) = \frac{1}{2\pi(\sigma_t^2 + \|\mathbf{v}_t^c\|\Delta t)} e^{-\frac{\|\mathbf{q}_t^0 + \mathbf{v}_t^c \Delta t - \mathbf{q}\|^2}{2(\sigma_t^2 + \|\mathbf{v}_t^c\|\Delta t)}} \qquad (7)$$

Let us examine the possible interrelations between $\mathbf{q}_t^m$ and $\mathbf{q}_{t+\Delta t}^m$. At time instant t the robot is still in the free space $\mathcal{C}_f$, i.e. the contact with an obstacle hasn't occurred yet.We want to choose $\mathbf{v}_t^c$ such that the position at time instant $t+\Delta t$, has certain desirable properties. The criterion we propose here for the choice of $\mathbf{v}_t^c$ is following:

> Choose $\mathbf{v}_t^c$ so that if the impact with an obstacle occurs during the move from $\mathbf{q}_t^m$ to $\mathbf{q}_{t+\Delta t}^m$ the probability that the resulting compliant motion will be either sticking in the goal or sliding towards the goal will be maximal.

If the probability that the "good" compliant motion will occur in the case of impact is considered as a probability that a plan for a given task succeeds, than the requirement above is simply asking for the maximization of the success probability. If we assume that $\mathbf{v}_t^c$ has constant intensity v we may express it in the form $\mathbf{v}_t^c = [v\cos\phi \quad v\sin\phi]^T$. The angle ϕ defines the direction of motion. The optimal direction of motion is the one that maximizes the probability of the success.

Having chosen the criterion we want to optimize, we need to express it mathematically. The probability of a success, $\Psi\{\text{success}\}$, can be expressed as a probability that the initial point $\mathbf{q}_t^m$ is in free space $\mathcal{C}_f$ and the end point $\mathbf{q}_{t+\Delta t}^m$ is inside the part of the obstacle $\mathcal{CB}_S(\mathbf{q}_t^m)$ that results in sliding motion towards the goal. The obstacle $\mathcal{CB}$ can be divided into two subsets

$$\mathcal{CB} = \mathcal{CB}_S(\mathbf{q}_t^m) \cup \mathcal{CB}_F(\mathbf{q}_t^m)$$

The set $\mathcal{CB}_S(\mathbf{q})$ is the set of configurations $\mathbf{q}' \in \mathcal{CB}$ such that if the motion starts in $\mathbf{q}$ and it is aimed towards $\mathbf{q}'$, the compliant motion after collision (which is inevitable since the path $\overline{\mathbf{q}\mathbf{q}'}$ intersects the obstacle boundary $\partial\mathcal{CB}$) results in either sticking in the goal or sliding towards the goal (S in the subscript of $\mathcal{CB}_S$ stands for "success"). Analogously, the set $\mathcal{CB}_F(\mathbf{q})$ is the set of configurations $\mathbf{q}' \in \mathcal{CB}$ such that, on the path from $\mathbf{q}$ towards $\mathbf{q}'$, the resulting compliant motion results either in sticking outside the goal or sliding away from it (the subscript F in $\mathcal{CB}_F$ stands for "failure").

Now the success probability $\Psi\{\text{success}\}$ can be written as

$$\Psi\{\text{success}\} = \Psi\{\mathbf{q}_t^m \in \mathcal{C}_f \wedge \mathbf{q}_{t+\Delta t}^m \in \mathcal{CB}_S(\mathbf{q}_t^m)\} \qquad (8)$$

Note that the set $\mathcal{CB}_S(\mathbf{q}_t^m)$ of "good" end points $\mathbf{q}_{t+\Delta t}^m$ depends on the point $\mathbf{q}_t^m$.

We already have distribution density functions of $\mathbf{q}_t^m$ and $\mathbf{q}_{t+\Delta t}^m$ (relations 6 and 7). If we substitute them in relation 8 we get

$$\Psi\{\text{success}\} = \int_{\mathbf{p}_i \in \mathcal{C}_f} \int_{\mathbf{p}_j \in \mathcal{CB}_S(\mathbf{p}_i)} \psi_{\mathbf{q}_t^m}(\mathbf{p}_i)\psi_{\mathbf{q}_{t+\Delta t}^m}(\mathbf{p}_j)d\mathbf{p}_i d\mathbf{p}_j$$

This integral is difficult to compute exactly. The method we have used is Monte-Carlo integration that drastically simplifies the complexity of calculating multidimensional integrals (see Appendix for a discussion of the exact method used).

Now we will comment on some benefits that this kind of set representation may have. Relations between obstacles and a robot in the configuration space are usually given algebraically as a set of constraints. However, the alternative method of representing objects in the world coordinates (and thus in the configuration space) is gaining more popularity. That is so-called bitmap representation where an object is represented as a set of "colored" pixels rather then a set of algebraic constraints, exemplified by a successful implementation of a path planning for many degrees of freedom in [1] and related work [6, 7]. The attractiveness of bitmap descriptions is twofold: firstly, raw sensor data tend to support bitmap representations more naturally and secondly, this approach encourages the applications of parallel computations.

The relation of analytical vs. Monte-Carlo integration techniques is more or less equivalent to the distinction between algebraic constraint and bitmap representations. The

simplicity in finding the success probability based on Monte-Carlo integration (and, implicitly, a bitmap representation) may be one of its advantages.

Figure 8 shows the results of a simulated peg-in-hole task planning process computed by this method. In this example, the velocity and duration of each move is constant; we need to choose the optimal direction that a move of such duration and velocity will take for the task to succeed. The start position is point A, and B, C, and D are the intermediate positions found by optimizing the direction at each step of the path generation. The three segments $(\overline{AB}, \overline{BC}, \overline{CD})$ are generated in three consecutive optimizations at points $A = \mathbf{q}_0^0$, B and C. The number of points used in the Monte-Carlo integration was 200. The other parameters had values $\sigma = 0.1, \sigma^m = 0.03, v = 1, \Delta t = 1$. This example shows that the continuous uncertainty models we have developed are capable of solving typical robot motion problems that have been traditionally solved using discrete techniques.

CONCLUSION

In this paper we have developed a new method for modeling uncertainties in robotic systems and demonstrated its applications in two cases. Our uncertainty model is based on stochastic differential equations and continuous probability distributions. All three types of uncertainties present in a robotic system — sensor, control and environment — can be modeled using the same principle, thus allowing the unified approach to planning of robot motions. We have implemented and experimented with two simple tasks that involve uncertainty: peg-in-hole insertion with a constant velocity in the presence of control uncertainty and planning of velocity profiles under time and force constraints in the presence of environment uncertainty. In both cases the criterion we optimized was the probability that the task will succeed in a sense that we have defined, and that seems intuitively reasonable. The method we have used for optimization of the success probability was analytical, demonstrating the applicability of these types of methods to problems where discrete search techniques have been utilized. Nevertheless, the quest for global extremum of an analytical function is genuinely a search process. It seems that the very nature of the planning problem requires a certain type of search procedure to take place, since in this case we have replaced search in a discrete space with search int he space of continuous analytic functions. However, there are indications that that replacement may lead to more efficient algorithms, specially in multidimensional or cluttered environments.

The general environment model is another topic that we consider as a contribution of this paper. Since all three types of uncertainties are encompassed in one unifying system of stochastic equations, the treatment of environment uncertainty is not any different than the treatment of other two types of uncertainty. Besides that, the recognition of the need for varying amounts of environment uncertainty allows for modeling the environments where the amount of knowledge changes as a function of a current position. We have demonstrated that the method of optimizing the success probability results in plans that are intuitively correct.

We hope that this research effort will result in an comprehensive system for planning tunable parameters of robotic tasks in the presence of uncertainty. There are however many issues that need to be addressed in the future work. The theoretical model developed in this paper shows promises that it can be used as a basis for the future system.

References

[1] J. Barraquand, B. Langlois, and J.-C. Latombe. Robot motion planning with many degrees of freedom and dynamic constraints. In *Robotics Research, the Fifth International Symposium*, pages 435–444, 1990.

[2] J. F. Canny and J. Reif. New lower bound techniques for robot motion planning problem. In *Proceedings of the 28th IEEE Symposium on Foundations of Computer Science*, pages 49–60, 1987.

[3] B. R. Donald. *Error Detection and Recovery in Robotics.* Springer-Verlag, 1987.

[4] H. F Durrant-Whyte. Uncertain geometry in robotics. In *Proceedings of the IEEE Conference on Robotics and Automation*, pages 851–856, 1987.

[5] M. Erdmann. Using backprojections for fine motion planning with uncertainty. *International Journal of Robotics Research*, 5(1), 1986.

[6] B. Faverjon and P. Tournassoud. A local based approach for path planning of manipulators with high number of degrees of freedom. In *Proceedings of the IEEE Conference on Robotics and Automation*, pages 1152–1159, 1987.

[7] B. Faverjon and P. Tournassoud. A practical approach to motion-planning for manipulators with many degrees of freedom. Technical report, INRIA, 1988.

[8] M. Friedlin. *Functional integration and partial differential equations.* Princeton University Press, 1985.

[9] G. D. Hager. *Task-Directed Sensor Fusion and Planning: A Computational Approach.* Kluwer Academic Publishers, 1991.

[10] E. Krotkov. *Active Computer Vision by Cooperative focusing and stereo.* Springer-Verlag, 1989.

[11] T. Lozano-Peréz. Spatial planning: A configuration space approach. *IEEE Transaction on Computers*, C-32(2):108–120, 1983.

[12] T. Lozano-Peréz, M. T. Mason, and R. H. Taylor. Automatic synthesis of fine-motion strategies for robots. *International Journal of Robotics Research*, 3(1):3–24, 1984.

[13] M. T. Mason. Compliance and force control for computer controlled manipulators. *IEEE Transaction on Systems, Man and Cybernetics*, SMC-1(6):418–432, 1981.

[14] R. McKendall and M. Mintz. Sensor-fusion with statistical decision theory: A prospectous of research in the grasp lab. Technical Report MS-CIS-90-68, GRASP Lab, University of Pennsylvania, 1990.

[15] D. E. Whitney. Force feedback control of manipulator fine motions. *ASME Journal of Dynamic Systems, Measurement and Control*, pages 91–97, 1977.

Derivation of the probability density $\psi_{\mathbf{q}_t^m}$

The probability density function $\psi_{\mathbf{q}_t^m}$ of the random variable $\mathbf{q}_t^m$ is given by a convolution-type integral

$$\psi_{\mathbf{q}_t^m}(\mathbf{q}) = \int_{-\infty}^{\infty} \psi_{\mathbf{q}_0^m}(\mathbf{q}') \Gamma_{\mathbf{q}_t^m}(\mathbf{q}', \mathbf{q}_t^c, \mathbf{q}) d\mathbf{q}' \qquad (9)$$

In the above relation $\psi_{q_0^m}(q)$ is the initial density at $t = 0$ in our case $\delta(\mathbf{q}^m - \mathbf{q}^c)$ since we have assumed that at $t = 0$ $\mathbf{q}^m$ and $\mathbf{q}^c$ coincide. $\Gamma_{\mathbf{q}_t^m}$ is the *transition density* which defines how $\psi_{\mathbf{q}_t^m}$ is being changed as $\mathbf{q}_t^c$ changes. The form of the integral in 9 resembles the form of a convolution integral that defines density of a sum of two random variables. This resemblance can be understood as a combination of two uncertainties: initial (which is, in this case, 0) and the one acquired over time and represented by $\Gamma_{\mathbf{q}_t^m}$.

Equation 1 is essentially n-dimensional system where n is the dimensionality of the configuration space $\mathcal{C}$. However, if we make the further assumption that σ^m is a diagonal matrix, different components of $\mathbf{q}_t^m$ become uncorrelated thus allowing us to consider 1 as a scalar equation. We will proceed by adopting this assumption and, accordingly, dropping boldface notation for q's.

It can be shown that the transition density $\Gamma_{q_t^m}(q', q_0^c, q)$ satisfies the backward Kolmogorov equation

$$\frac{\partial}{\partial q_0^c}\Gamma_{q_t^m} = \frac{\partial}{\partial q'}\Gamma_{q_t^m} + \frac{1}{2}\frac{\partial^2}{\partial q_0^{c2}}\Gamma_{q_t^m} \tag{10}$$

with an initial condition

$$\lim_{q_t^c \to q_0^c} \Gamma_{q_t^m} = \delta(q - q') \tag{11}$$

The limit in the initial condition can be substituted with more frequently used $\lim_{t\to 0}$ (since when $t \to 0$ we have $q_t^c \to q_0^c$).

The solution of equation 10 with initial condition 11 is given by

$$\Gamma_{q_t^m}(q', q_0^c, q) = \frac{1}{\sqrt{2\pi}\sigma^m\sqrt{q_t^c - q_0^c}} e^{-\frac{(q-q'-(q_t^c-q_0^c))^2}{2\sigma^{m\,2}(q_t^c-q_0^c)}} \tag{12}$$

Now by substituting $\Gamma_{q_t^m}$ and $\psi_{q_0^m}$ into the relation 9 we obtain the expression 3 for $\psi_{q_t^m}$.

Monte-Carlo Integration of the Success Probability

Let $\mathcal{P} = \{\mathbf{p}_i | i = 1, \ldots, N\}$ be the set of random points uniformly distributed in the configuration space $\mathcal{C}$. Furthermore, let $\mathcal{P}_f = \mathcal{P} \cap \mathcal{C}_f$ and $\mathcal{P}_S(\mathbf{p}) = \mathcal{P} \cap \mathcal{CB}_S(\mathbf{p})$. Using the formula for the Monte-Carlo integration, the success probability integral can be rewritten as

$$\Psi\{\text{success}\} = \frac{\|\mathcal{C}_f\|}{\|\mathcal{P}_f\|} \sum_{\mathbf{p}_i \in \mathcal{P}_f} \frac{\|\mathcal{CB}_S(\mathbf{p}_i)\|}{\|\mathcal{P}_S(\mathbf{p}_i)\|} \times \sum_{\mathbf{p}_j \in \mathcal{P}_S(\mathbf{p}_i)} \psi_{\mathbf{q}_t^m}(\mathbf{p}_i)\psi_{\mathbf{q}_{t+\Delta t}^m}(\mathbf{p}_j) \tag{13}$$

where $\|\mathcal{C}_f\|$ and $\|\mathcal{CB}_S(\mathbf{p}_i)\|$ are surfaces and $\|\mathcal{P}_f\|$ and $\|\mathcal{P}_S(\mathbf{p}_i)\|$ are cardinalities of appropriate sets.

Under the assumption that the points in $\mathcal{P}$ are uniformly distributed over $\mathcal{C}$ we approximately have $\|\mathcal{C}_f\|/\|\mathcal{P}_f\| = \|\mathcal{CB}_S(\mathbf{p}_i)\|/\|\mathcal{P}_S(\mathbf{p}_i)\| = \|\mathcal{C}\|/\|\mathcal{P}\|$. Thus, the expression 13 reduces to

$$\Psi\{\text{success}\} = \left(\frac{\|\mathcal{C}\|}{N}\right)^2 \sum_{\mathbf{p}_i \in \mathcal{P}_f} \sum_{\mathbf{p}_j \in \mathcal{P}_S(\mathbf{p}_i)} \psi_{\mathbf{q}_t^m}(\mathbf{p}_i)\psi_{\mathbf{q}_{t+\Delta t}^m}(\mathbf{p}_j)$$

Now we need to find ϕ that maximizes Ψ. Since Ψ is differentiable in ϕ we may look for the zero of the equation $\partial\Psi/\partial\phi = 0$. This equation can be solved by Newton's method using the iteration

$$\phi^{(\nu+1)} = \phi^{(\nu)} - \frac{\partial\Psi/\partial\phi}{\partial^2\Psi/\partial\phi^2}$$

The complexity of the algorithm for computing Ψ is $O(N^2)$ where N is total number of points in Monte-Carlo integration.