# Did the particles hit the wall? A particle based method to robust control

**Uwe Schmidt**
Department of Computer Science
University of British Columbia
Vancouver, B.C.
*uschmidt@cs.ubc.ca*

## Abstract

This paper is about the implementation and details of a particle control method by Lars Blackmore. It achieves optimal robust predictive control by approximating all sources of uncertainty with particles to transform a stochastic control problem into a deterministic one. This can be solved to global optimality using Mixed Integer Linear Programming (MILP) in the case of linear system dynamics and a linear cost function. This is an interesting approach because it can be used for robust vehicle path planning under uncertainty without relying on Gaussian distributions. I will first give an overview over the method and then present details on how to model vehicle path planning as a MILP. My results verify the feasibility of Blackmore's method.

## 1 Introduction

I chose to implement the paper *A Probabilistic Particle Control Approach to Optimal, Robust Predictive Control* [1] by Lars Blackmore. Blackmore introduces a new method for approximating a stochastic control problem with a deterministic one. This is done by approximating all probability distributions with particles.

Predictive stochastic control takes into account uncertainty in dynamic systems to control the system state over a certain planning horizon and *Robust* stochastic control can additionally ensure that the probability of failure is below a specified threshold (called a chance constraint). Robust path planning for vehicles under uncertainty is an important application for this method. Failure can be defined in terms of colliding with an obstacle or not reaching a goal area. *Optimal* robust predictive stochastic control tries to find the best control sequence by minimizing a given cost function. Optimal robust predictive control is a very hard problem since it must optimize over the space of possible future states. Previous approaches either attacked the problem by discretizing the state space and assuming only a finite number of actions in each state, or they use constrained optimization techniques to find a solution. The latter methods require a linear dynamic system model and all sources of uncertainty to be Gaussian. Blackmore's contribution is the development of a new method which approximates all sources of uncertainty in the stochastic control problem with particles, transforming it into a deterministic control problem. Then he focuses on the case of linear dynamic systems and a linear cost function, since it is possible to solve the constrained optimization problem to global optimality using Mixed Integer Linear Programming (MILP) in this case. His method also takes chance constrains into account, thus enabling robust control.

The bulk of my time in this project was spent on the implementation of the method as described in his paper. My contributions in this report are mainly that I give details about how to model vehicle path planning as a MILP in section 4 and show my results in section 6. Unless otherwise indicated, all content in this paper is taken from [1].

## 2 Problem Statement

> *"Design a finite, optimal sequence of control inputs, taking into account probabilistic uncertainty, which ensures that the state of the system leaves a given feasible region with probability at most $\delta$."* [1]

Uncertainty is given by a probabilistic distribution over the initial state and by disturbances which act on the system state at each time step. Additionally the system model may not be known exactly. The form of these distributions is not restricted as long as it is possible to sample from them.

## 3 Method

Initially we have a stochastic control problem with the objective to minimize the cost function $h(u_0, \dots, u_{T-1}, x_{1:T})$ while assuring that the state trajectory $x_{1:T}$ leaves a given feasible region $F$ with probability at most $\delta$: $p(x_{1:T} \notin F) \leq \delta$. The following steps explain how to approximate the stochastic problem with a deterministic one and how to solve it to global optimality while obeying the given chance constraint.

1. Generate $N$ samples $\{x_0^{(1)}, \dots, x_0^{(N)}\}$ from the initial state distribution and $T \cdot N$ samples $\{\nu_0^{(1)}, \dots, \nu_0^{(N)}, \dots, \nu_{T-1}^{(1)}, \dots, \nu_{T-1}^{(N)}\}$ from the disturbance process.
2. The approximated distribution of future state trajectories is dependent on the samples generated in step 1 and the control inputs $u_0, \dots, u_{T-1}$. Each particle $x_{1:T}^{(i)}$ represents a state trajectory over the entire planning horizon and can be expressed in the following way:
$$x_{1:T}^{(i)} = \begin{bmatrix} x_1^{(i)} & \dots & x_T^{(i)} \end{bmatrix}^T \quad x_t^{(i)} = f_t(x_0^{(i)}, u_0, \dots, u_{t-1}, \nu_0^{(i)}, \dots, \nu_{t-1}^{(i)}) \tag{1}$$
3. Approximate the probability of failure by the the distribution of particles:
$$p(x_{1:T} \notin F) \approx \frac{1}{N} |\{x_{1:T}^{(i)} \notin F, i = 1, \dots, N\}| \tag{2}$$
4. Approximate the cost function in terms of particles:
$$h(u_0, \dots, u_{T-1}, x_{1:T}) \approx \hat{h}(u_0, \dots, u_{T-1}, x_{1:T}^{(1)}, \dots, x_{1:T}^{(N)}) \tag{3}$$
5. Solve the deterministic constrained optimization problem:
$$\begin{array}{ll} \min & \hat{h}(u_0, \dots, u_{T-1}, x_{1:T}^{(1)}, \dots, x_{1:T}^{(N)}) \\ \text{subject to} & \frac{1}{N} |\{x_{1:T}^{(i)} \notin F, i = 1, \dots, N\}| \leq \delta \end{array}$$

The optimal solution for the control inputs $u_0, \dots, u_{T-1}$ obtained in step 5 above will give us the solution to the problem posed in section 2.

## 4 Modelling Vehicle Path Planning

This section provides details on how to model robust vehicle path planning as a MILP using the method in section 3 in the case of linear system dynamics and a linear cost function.

The formal definition of a MILP [2]: $\min_{x,z}\ c_1^T x + c_2^T z,$ subject to $A_1^T x + A_2^T z \leq b$, where $x$ are continuous and $z$ are integer variables, $c$ defines the cost function and $A$ is a matrix used to encode constraints on $x$ and $z$. A special and important case of integer variables are binary variables which only take values 0 and 1 because they enable us to encode logic. Using an example from [2], suppose we have the following constraints:

$$\begin{aligned} a_1^T x &\leq b_1 + M z_1 && (4)\\ a_2^T x &\leq b_2 + M z_2 && (5)\\ z_1 + z_2 &\leq 1,\ \ z_i \in \{0,1\} && (6) \end{aligned}$$

The constraint in equation (6) forces $z_1$ *or* $z_2$ to be 0, or both of them. If we assume that $M$ is a big positive constant with the property $M \geq \max\{a_1^T x - b_1, a_2^T x - b_2\}$, then (4) is always true if $z_1 = 1$ and (5) is always true if $z_2 = 1$. This means that only one of the constraints $a_1^T x \leq b_1$ *or* $a_2^T x \leq b_2$ must be true (disjunction of constraints) to satisfy all constraints (4)–(6). This technique is known as the *big-M* method and will be used throughout this paper to encode logic constraints. All binary variables in the remainder of this paper have a value range of $\{0, 1\}$.

### 4.1 Linear System Model

We require a linear discrete time system model of the form $x_{t+1} = Ax_t + Bu_t + \nu_t$ because we can only encode linear constraints in a MILP. All future values of $x_t^{(i)}$ solely depend on the initially sampled $x_0^{(i)}$ and $\nu_j^{(i)}$. The system model in the previous sentence gives us the function $f_t$ used in equation (1) so we can express the future particle states as follows:

$$x_t^{(i)} = A^t x_0^{(i)} + \sum_{j=0}^{t-1} A^{t-j-1}(Bu_j + \nu_j^{(i)})\ {}^{1} \qquad (7)$$

Blackmore uses aircraft models to demonstrate the feasibility of his new method. I considered a simpler two-dimensional linear vehicle model:

$$\begin{aligned} x_{t+1} &= x_t + d \cdot v_t + \nu_t && (8)\\ v_{t+1} &= cons \cdot v_t + tr \cdot d \cdot u_t && (9) \end{aligned}$$

where $d$ is the discretization step, $cons$ is the "velocity conservation" and $tr$ is the "transmission" of the control input $u_t$. I constrained the velocity $|v_t| \leq Vmax$ and the control input $|u_t| \leq Umax$ and additionally put limits on the amount of change in $u_t$ at each time step ($u_t - \alpha \leq u_{t+1} \leq u_t + \alpha$).

### 4.2 Robustness

The number of particles that fall outside of the feasible region have to be constrained to obey the given chance constraint. For this reason, binary variables $\{z_i\}_{i=1}^N$ are introduced which are used to indicate whether a given particle lies inside the feasible region or not. We want to have the property that a value of $z_i = 0$ implies $x_{1:T}^{(i)} \in F$. The sum over all $z_i$ then gives us the number of failing particles, which must not be greater than $\delta N$: $\frac{1}{N}\sum_{i=1}^{N} z_i \leq \delta$. How to set constraints to accomplish the desired property $z_i = 0 \Leftrightarrow x_{1:T}^{(i)} \in F$ is discussed in the following subsections.

[1]Please note that this equation appears with a typo in Blackmore's paper (equation 18).

### 4.3 Arrival

Blackmore mentions that failure can also be defined in terms of failing to arrive at a goal area — but never gives the details about it. I present a possible way how to do it. I introduced binary variables $a_{it}$ with the desired property that $a_{it} = 1$ implies that particle $i$ is in the goal area at time $t$. The sum over all $a_{it}$ for a each particle is then constrained:

$$\forall i \in \{1, \ldots, N\} : \sum_{t=1}^{T} a_{it} \geq 1 - M z_i \tag{10}$$

This constraint guarantees that particle $i$ has reached the target at some time step if $z_i = 0$. A value of $z_i = 1$ leaves particle $i$ unconstrained since equation (10) will be always true if we assume $M$ to be a big postive constant. One could also extend this to visit multiple way-points.

I considered a rectangular goal area $G = \{x_{min}, x_{max}, y_{min}, y_{max}\}$ and need therefore four constraints for $x_t^{(i)} = \left[x_{t_1}^{(i)}\ x_{t_2}^{(i)}\right]^T$ which have to be met to ensure arrival: $x_{t_1}^{(i)} - x_{min} \geq M(a_{it} - 1)$, $x_{t_2}^{(i)} - y_{min} \geq M(a_{it} - 1)$, $x_{t_1}^{(i)} - x_{max} \leq M(1 - a_{it})$, and $x_{t_2}^{(i)} - y_{max} \leq M(1 - a_{it})$.

These constraints ensure that $x_t^{(i)}$ is in the target area $G$ if $a_{it} = 1$, but leaves $x_t^{(i)}$ unconstrained otherwise. This gives us the desired property for equation (10).

### 4.4 Linear Cost Function

The cost function $h$ can be based on the control inputs $u_t$ and binary decision variables. The used cost function $\hat{h}$ can only be an approximation of the original cost function $h$ since we are approximating all probability distributions with particles.

I chose the following linear cost function ($u_t = [u_{t_1}\ u_{t_2}]^T$):

$$\hat{h} = w_a \left[\sum_{i=1}^{N} \sum_{t=1}^{T} t \cdot a_{it}\right] + w_u \left[\sum_{t=1}^{T-1} |u_{t_1}| + |u_{t_2}|\right] \tag{11}$$

which punishes the control effort $|u_{t_1}| + |u_{t_2}|$ and late arrival times at the target. The weights $w_a$ and $w_u$ can be chosen so that either an early arrival time or minimizing the magnitude of control is more important.

One problem is to base the cost function directly on the magnitude of the control variables $u_t$, since the absolute value function is not linear. Blackmore doesn't mention how to do this in his paper. I found out that it can be done by introducing additional continuous variables $m_t$ and putting the following constraints for each time step $t$: $m_t \geq u_{t_1} + u_{t_2}$, $m_t \geq u_{t_1} - u_{t_2}$, $m_t \geq -u_{t_1} + u_{t_2}$, and $m_t \geq -u_{t_1} - u_{t_2}$.

These constraints are based on the following observation:

$$|u_{t_1}| + |u_{t_2}| = \max\{u_{t_1} + u_{t_2}, u_{t_1} - u_{t_2}, -u_{t_1} + u_{t_2}, -u_{t_1} - u_{t_2}\} \tag{12}$$

This ensures that $m_t \geq |u_{t_1}| + |u_{t_2}|$ for each time step and basing the cost function on $m_t$ minimizes over $|u_{t_1}| + |u_{t_2}|$.

### 4.5 Non-convex Feasible Regions

Vehicle path planning in the case of obstacle avoidance can be done with non-convex polygonal feasible regions. Instead of being inside a convex feasible region, the vehicle has to avoid several convex *in*feasible regions (obstacles) at each time step. Each obstacle can be approximated by a convex polygon and is avoided if the vehicle is "outside" any of the obstacle's edges. A single obstacle is avoided if one of the linear constraints $a_{jl}^T x_t \geq b_{jl}$ (figure 1) is satisfied (disjunction of constraints). The following constraints are used by Blackmore to enable obstacle avoidance with a probability of failure of at most $\delta$: $a_{jl}^T x_t^{(i)} - b_{jl} \geq -M d_{ijtl}$, $\sum_{l=1}^{N_j} d_{ijtl} - (N_j - 1) \leq M e_{ijt}$, $\sum_{t=1}^{T} e_{ijt} \leq M g_{ij}$, and $\sum_{j=1}^{L} g_{ij} \leq M z_i$.

The basic idea is to integrate over "one dimension" of constraints at each step. The binary variables $d_{ijtl}$, $e_{ijt}$ and $g_{ij}$ are introduced for this purpose.

The first of the above constraints ensures that particle $i$ satisfies constraint $l$ of obstacle $j$ at time $t$ if $d_{ijtl} = 0$. To avoid obstacle $j$ at time $t$, particle $i$ has to satisfy at least one of the $N_j$ constraints (i.e. at least one $d_{ijtl} = 0$) which is indicated by $e_{ijt}$. Whether particle $i$ avoids obstacle $j$ at all time steps is expressed with $g_{ij}$ and finally $z_i$ indicates whether all obstacles are avoided by particle $i$ (i.e. particle $i$ is inside the feasible region). Overall, we first integrate over all obstacle constraints, then over all time steps, and finally over all obstacles to have a single variable $z_i$ to indicate the feasibility of $x_{1:T}^{(i)}$.

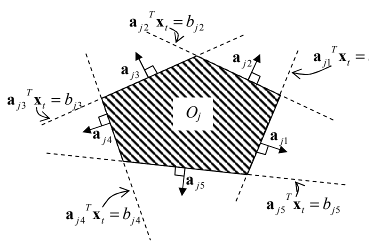
$\mathbf{a}_{j2}^T \mathbf{x}_t = b_{j2}$
$\mathbf{a}_{j1}^T \mathbf{x}_t = b_{j1}$
$\mathbf{a}_{j3}$
$\mathbf{a}_{j2}$
$\mathbf{a}_{j3}^T \mathbf{x}_t = b_{j3}$
$O_j$
$\mathbf{a}_{j4}$
$\mathbf{a}_{j1}$
$\mathbf{a}_{j5}$
$\mathbf{a}_{j5}^T \mathbf{x}_t = b_{j5}$
$\mathbf{a}_{j4}^T \mathbf{x}_t = b_{j4}$

Figure 1: An example of a non-convex feasible region (obstacle). The obstacle is defined by $N_j$ line segments and their outwards pointing normal vectors $a_{jl}$. Source: [1]

## 5 Implementation

I implemented the algorithm in MATLAB in overall more than 400 lines of code, which includes all steps described in section 4. Additionally, I wrote a function which computes all line segments and outwards pointing normal vectors of a polygonal obstacle, given the points in clockwise or counterclockwise order. The following results only use a few particles because the number of binary variables is mainly determined by the number of particles and grows very fast. I am using the GNU Linear Programming Kit (GLPK)[2] and GLPKMEX[3] as an interface to MATLAB to solve the MILP. GLPK is an open-source solver which is presumably slower and less efficient than commercial solver packages like CPLEX[4].

## 6 Results

I ran a simulation over $T = 13$ time steps with $N = 5$ particles using a Gaussian distribution for the initial state and state disturbances: $\nu_t^{(i)} \sim x_0^{(i)} \sim \mathcal{N}(0, 1/250)$. The following parameters were used for the vehicle model described in section 4.1: $v_0 = u_0 = 0$, $Vmax = Umax = 1$, $\alpha = Ulim/5$, $tr = cons = 0.9$ and $d = 0.5$. I used the cost function described in section 4.4 with the weights $w_a = w_u = 1/N$, thus emphasizing early arrival times over control effort. Figure 2 shows the path of particles with a probability of failure of $\delta = 0.4$ (i.e. 2 out 5 particles are allowed to collide with an obstacle

[2]`http://www.gnu.org/software/glpk/`

[3]`http://www.dii.unisi.it/~hybrid/tools/mex/downloads.html`

[4]`http://www.ilog.com/products/cplex`

and not to reach the goal). The cost for this solution is 3.0028 with a control magnitude of $\sum_{t=1}^{T-1} |u_{t_1}| + |u_{t_2}| = 4.3978$.

In the same scenario, with a lower probability of failure of $\delta = 0.2$, it is not possible to find a control sequence through the narrow path that ensures this level of conservatism. Thus the longer path around the top obstacle is chosen (figure 3). This "safe" path has a cost of 8.8070 with a control magnitude of $\sum_{t=1}^{T-1} |u_{t_1}| + |u_{t_2}| = 5.7585$. The cost is much higher because of the arrival component of the cost function. The difference in control magnitude shows that conservatism comes at the expense of control effort. In summary, I could verifiy the feasibility of the method described in section 3 in my simulations.

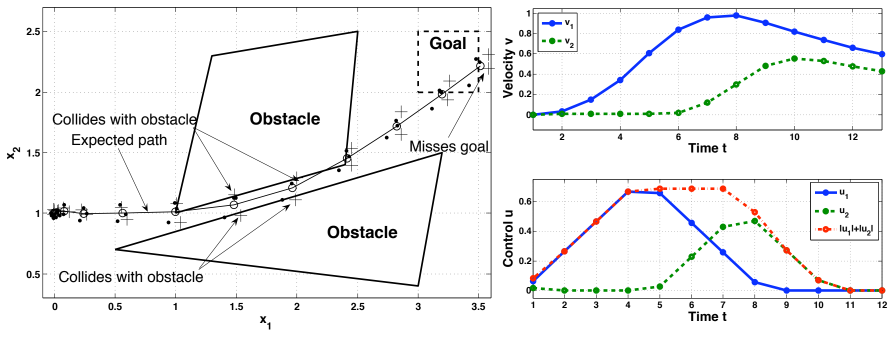

Figure 2: The vehicle takes the path through the narrow corridor for $\delta = 0.4$. It can be seen that exactly 2 particles (shown as $+$) collide with the obstacles and one of them additionally doesn't reach the goal.

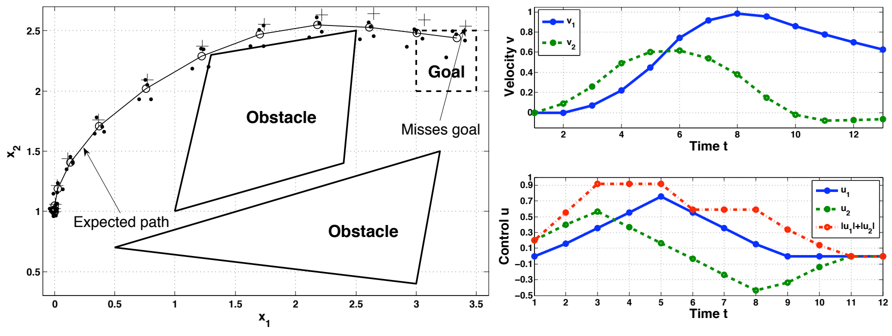

Figure 3: The vehicle takes the longer path around the top obstacle because of a more conservative probability of failure of 0.2. It can also be seen that a single particle doesn't reach the goal.

**Acknowledgement**: I would like to thank Emtiyaz and Lucas for their help during this project.

## References

[1] L. Blackmore. A Probabilistic Particle Control Approach to Optimal, Robust Predictive Control. *Proceedings of the AIAA Guidance, Navigation and Control Conference*, 2006.

[2] A. Richards and J. How. Mixed-integer programming for control. *Proceedings of the American Control Conference*, 2005.