Model-Predictive Control with Stochastic Collision Avoidance using Bayesian Policy Optimization

Olov Andersson[1], Mariusz Wzorek[1], Piotr Rudol[1] and Patrick Doherty[1]

Abstract— **Robots are increasingly expected to move out of the controlled environment of research labs and into populated streets and workplaces. Collision avoidance in such cluttered and dynamic environments is of increasing importance as robots gain more autonomy. However, efficient avoidance is fundamentally difficult since computing safe trajectories may require considering both dynamics and uncertainty. While heuristics are often used in practice, we take a holistic stochastic trajectory optimization perspective that merges both collision avoidance and control. We examine dynamic obstacles moving without prior coordination, like pedestrians or vehicles. We find that common stochastic simplifications lead to poor approximations when obstacle behavior is difficult to predict. We instead compute efficient approximations by drawing upon techniques from machine learning. We propose to combine policy search with model-predictive control. This allows us to use recent fast constrained model-predictive control solvers, while gaining the stochastic properties of policy-based methods. We exploit recent advances in Bayesian optimization to efficiently solve the resulting probabilistically-constrained policy optimization problems. Finally, we present a real-time implementation of an obstacle avoiding controller for a quadcopter. We demonstrate the results in simulation as well as with real flight experiments.**

I. INTRODUCTION

As robots gain increased autonomy, they are expected to safely navigate populated environments and work alongside humans in the workplace. Safe collision avoidance is therefore of central importance for robots to be accepted into wider society. Efficient collision avoidance is generally considered a difficult problem as one may have to take both dynamics and uncertainty into account. The last two decades have seen considerable research into this area [1], but collision avoidance in unmanaged real-world environments remains difficult. In this paper we take a holistic approach to the safe collision avoidance problem involving humans and derive approximate solutions using machine learning and fast model-predictive control solvers.

Most prior work on moving obstacles make use of strong assumptions on dynamics, uncertainty or cooperative behavior. Velocity obstacles [2] is a popular heuristic that ignores higher order dynamics, uncertainty and assumes obstacles follow a predetermined trajectory. Many authors have relaxed some of these assumptions, for example to include state uncertainty [3], unconstrained linear dynamics [4], reciprocal obstacle behavior [5] or a combination thereof [6]. Humans can however be unpredictable, inattentive and can not necessarily be counted on to be cooperative.

To accurately take both dynamics and uncertainty into account, we frame the problem as stochastic trajectory optimization. This allows a principled way to solve both control and collision avoidance in the same framework. Previous work mainly considers the case of static obstacles [7] or multi-robot scenarios where the control policies of other robots are known [8].

To achieve safety we relax cooperative assumptions in prior work and model humans as non-cooperative moving obstacles under uncertainty, effectively giving them a right-of-way privilege. We show that for non-cooperative moving obstacles, common uncertainty assumptions result in poor approximations of the safety constraints. We identify the weakness as a deficiency in the modeling of controller recourse, the capacity to adapt to obstacle behavior.

However, accurate modeling of controller recourse is difficult in continuous domains. Instead we propose an approximation by drawing upon techniques from machine learning. Policy search [9] are techniques from reinforcement learning that can provide useful approximations to stochastic control problems. Their drawback is that they rely on expert selection of a policy parameterization suitable for the task. These range from linear control laws to more complex motion primitives [10].

We propose that the output of a model-predictive controller (MPC) with a parameterized constraint function can be used as a control policy. This allows us to approximate difficult stochastic control problems with fast deterministic solvers such as [11]. By using parameterized soft safety constraints, we can find a controller producing safe trajectories for a stochastic target domain with a chosen level of confidence. Optimizing these policy parameters is a much smaller but still difficult probabilistically-constrained stochastic optimization problem in itself. We further propose to solve this using recent advances in constrained Bayesian optimization [12][13].

The contributions of this paper are threefold. First, we consider common assumptions for stochastic trajectory optimization and show that they lead to poor approximations for non-cooperative moving obstacles. Second, we draw upon techniques from machine learning to find a more pragmatic approximation. We combine recent fast constrained MPC solvers with the stochastic properties of policy search. An efficient solution is found using Bayesian policy optimization. Finally, we present a model-predictive controller capable of

[1]Dept. of Computer and Information Science, Linköping University, SE-58183, Linköping, Sweden. This work was supported by grants from the Swedish Research Council (VR) Linnaeus Center CADICS, the ELLIIT Excellence Center at Linköping-Lund for Information Technology, the Swedish Aeronautics Research Council (NFFP6), Swedish Foundation for Strategic Research (SSF) project CUAS, and the EU project SHERPA.

real-time collision avoidance on a quadcopter. The results are demonstrated in simulations and using real flights.

The remainder of this paper is organized as follows. In section II we introduce the stochastic trajectory optimization perspective used to solve the collision avoidance problem. In section III we introduce models of non-cooperative moving obstacles like humans, as well as problems pertaining to common stochastic assumptions. In section IV we formulate the problem as probabilistically constrained policy search, using a policy parameterized by a model-predictive control solver. We propose an efficient solution to this by using Bayesian optimization. In section V we introduce a real-time capable collision avoiding quadcopter controller. Finally, we present the results of multiple experiments with the quadcopter in section VI.

II. STOCHASTIC TRAJECTORY OPTIMIZATION

Consider a robot with the state vector $\mathbf{x} \in \mathbb{R}^n$, control vector $\mathbf{u} \in \mathbb{R}^m$ and transition dynamics $\mathbf{x}_t = f(\mathbf{x}_{t-1}, \mathbf{u}_{t-1})$. By solving a discrete-time optimization problem, trajectory optimization aims to select the future controls $\mathbf{u}_{0..T-1}$ that generates a trajectory $\mathbf{x}_{1..T}$ with minimal cost $c(\mathbf{x}_{1..T}, \mathbf{u}_{0..T-1})$. In many practical applications, constraint (vector) functions $\mathbf{g}(\mathbf{x}_t, \mathbf{u}_{t-1}) \geq \mathbf{0}$ also need to be observed along the trajectory. These can include control saturations, speed limits, or geometric constraints to enable collision avoidance.

Model-predictive control is an online application of trajectory optimization where at each time step a trajectory with fixed planning horizon T is computed, typically 10-100 time steps, and only $\mathbf{u}_0$ is used. In our experiments we use 40 steps (10 Hz). While theoretically sub-optimal, in many cases there is little gain with a longer horizon. Fast MPC solvers are increasingly available, in particular for convex problems [14][11]. These have cubic complexity in the number of constraints per time step ($\gtrsim n$), and interior-point solvers like [11] have linear time complexity in the planning horizon.

Stochastic trajectory optimization, shown in Eq. (1), extends deterministic trajectory optimization to domains with uncertainty in the dynamics or state estimation. Formally, trajectories evolve according to a probability distribution $f(\mathbf{x_t}|\mathbf{x}_{t-1}, \mathbf{u}_{t-1})$ from some prior $p(\mathbf{x}_0)$. We now compute an *expected* cost over these, and most importantly, we need *probabilistic* constraints with confidence p. This is equivalent to a continuous domain Markov decision process with constraints, possibly under partial observability.

$$\begin{aligned} \underset{\mathbf{u}_0 \dots \mathbf{u}_{T-1}}{\text{arg min}} \quad & \mathbb{E}\left[\sum_{t=1}^{T} c(\mathbf{x}_t, \mathbf{u}_{t-1})\right] \\ \text{subject to} \quad & \\ & \mathrm{p}(\mathbf{g}(\mathbf{x}_t, \mathbf{u}_{t-1}) \geq \mathbf{0}) > p, \\ & \text{where } t = 1, \dots, T. \end{aligned} \tag{1}$$

The general constrained non-linear probabilistic case is difficult and rarely feasible to solve in anything approaching real-time. For linear-Gaussian problems, uncertainty can be propagated in closed-form using a Kalman filter, or approximated by such. This also allows easier approximations for some probabilistic constraints, e.g. [15],[8]. For the special case of unconstrained linear-quadratic Gaussian systems, a deterministic solver using the linear-Gaussian mean estimate is also stochastically optimal. Belief space augmentations are also possible for linear-Gaussian approximations, which allows planning with regard to information acquisition. However that typically scales as $O(n^6)$ in the state space [4].

A. Obstacle Constraints

Obstacle avoidance can be modeled as geometric constraints on each time step of the trajectory. A simple formulation is to constrain the distance between the position of the robot and the obstacle, $\mathrm{dist}(\mathbf{p}_{r,t}, \mathbf{p}_{o,t}) = \|\mathbf{p}_{r,t} - \mathbf{p}_{o,t}\|$, to be positive. Here $\mathbf{p}_{r,t}$ is a subspace of $\mathbf{x}_t$, and uncontrollable obstacle state like $\mathbf{p}_{o,t}$ can be treated as constants in the optimization. Obstacle constraints are typically non-convex, and while they can be solved by standard methods like sequential quadratic programming (SQP) [16], we propose an alternative projection approach in section V. However, when the obstacles are moving without prior coordination, the problem will be stochastic and the constraints will be probabilistic against predicted obstacle position. For linear-Gaussian approximations, distance constraints have closed form solutions based on the confidence ellipsoid of a multivariate normal distribution. Although the proposed method is not limited to this, for efficiency and simplicity, in this paper we use isometric covariances by taking the max over dimensions, $\mathrm{var}_{\max}(\mathbf{p}_r - \mathbf{p}_o)$. We can then use distance constraints of the form

$$\mathbb{E}[\mathrm{dist}(\mathbf{p}_{\mathrm{r,t}}, \mathbf{p}_{\mathrm{o,t}})] > \sqrt{\chi^2(1-p)\,\mathrm{var}_{\max}(\mathbf{p}_{r,t}, \mathbf{p}_{o,t})}. \tag{2}$$

Assuming, the closest points on the robot and obstacle can be found, the desired degree of confidence p can be realized with χ^2-distribution with k degrees of freedom in $\mathbb{R}^k$. However, since the confidence is per constraint and instant in time, and these are not independent, it is difficult to set the overall confidence level. More advanced online risk allocation strategies have been proposed [17], but they add complexity and are difficult to apply to model-predictive control where only a short planning horizon is used. In this paper we will use simple approximations like in Eq. (2) and instead propose a method to calibrate them. That way we can reach a desired level of safety confidence on more intuitive criteria, like minutes of scenario time.

III. HUMAN OBSTACLE MODELS

As robots are leaving lab environments to autonomously navigate streets or share work environments, the most pressing safety issue is avoiding collision with humans. While attentive humans subtly cooperate with each other to some extent while navigating, it is unclear how far that cooperation extends to objects like robots that do not share the same size, modalities or legal rights. Small robots might be stepped on, and the rotors on a quadcopter could cause injury. To ensure safety, a conservative assumption is therefore to consider humans non-cooperative and always give them right-of-way.

Given a stochastic model of human movement, we could bring the framework of stochastic trajectory optimization from section II to bear on finding safe trajectories even if humans change course without concern for the robot.

One popular model of movement is to use a random Gaussian acceleration model. This model has previously been used for moving obstacles in simulation studies like [18].

A random Gaussian model would also have practical advantages. A Kalman filter could compute the predictive distribution of the obstacle. Assuming the robot dynamics and state uncertainty was also approximately linear-Gaussian, it would be feasible to compute probabilistic constraints in closed form similarly to Eq. (2).

However, when we attempt to construct a realistic motion model using random Gaussian acceleration, we run into some problems. Here we focus on a motion model for walking humans. While there has been considerable research on high level human behavior, there is less on the level of dynamics and realistic validated stochastic models of individual humans. We use the model from [19], where the max velocity of a walking human is estimated to be $1.29\,\mathrm{m\,s^{-1}}$. For simplicity, their acceleration profile can be closely bounded by a constant acceleration of $1\,\mathrm{m\,s^{-2}}$.

To translate this into a random acceleration model we set the standard deviation of the random acceleration to $1\,\mathrm{m\,s^{-2}}$. This will surpass the profile some of the time, but it is also optimistic in that it assumes independence. In Fig. 1 we plot prediction intervals $4\,\mathrm{s}$ into the future, the trajectory planning horizon we use in this paper. The pedestrian starts with a forward velocity of $1.29\,\mathrm{m\,s^{-1}}$. As seen, this quickly grows into extremely wide intervals, even at the 95% level.

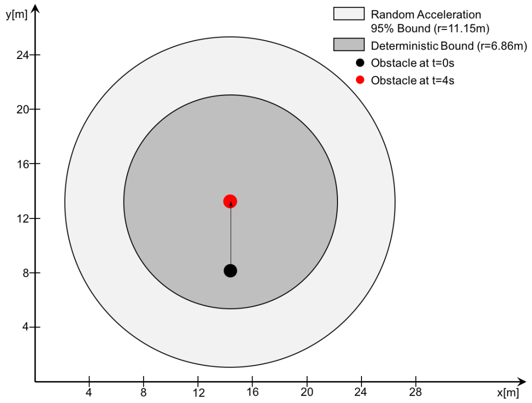


Fig. 1: Human prediction safety margins.

Humans are very agile, and may be capable of sustaining such acceleration for a prediction horizon of $4\,\mathrm{s}$, but it is not a reasonable model of a pedestrian. The intent with the human obstacle model is to be conservative but not adversarial. We need to impose the estimated max walking velocity from [19], but then the problem will not be linear-Gaussian, we cannot use a Kalman filter and the stochastic optimization problem will be more difficult.

An alternative obstacle model is to use a deterministic worst-case bound, by assuming the pedestrian turns and walks towards the robot. This bound is also used as a safety margin distance in Fig. 1. As a reference, the distance a pedestrian will cover from rest, is $4.3\,\mathrm{m}$, still a large number. Deterministic bounds have previously been used in e.g. [20], but they are still pessimistic and do not fit well with the stochastic optimization view of robot and state uncertainty.

The second and main problem why these are so unrealistic is that this is a constraint on the *predictive* obstacle distribution from this point in time only. If a robot was to plan a safe trajectory using predictive distributions only, it would always need to plan to end its trajectory 6 or 11 meters away, since initially it knows little of how the obstacle will move.

However, in practice, if the obstacle changes direction from the mean prediction, that prediction is updated and the controller has *recourse*, it will try to adapt. The predicted future control inputs $\mathbf{u}_{0..T-1}$ are not static but can change as the obstacle adjusts its course. Probabilistically, future controls are not independent of future state. Ignoring recourse means relying entirely on pessimistic predictive distributions such as above, and for highly mobile obstacles like humans, this also leads to very pessimistic controllers.

While recourse can be captured in global policy methods, classically based on dynamic programming [21], unlike trajectory optimization they scale poorly to high-dimensional problems with dynamics. Trajectory optimization approaches based on linear quadratic regulators or differential dynamic programming [22], e.g. [4], also has local linear recourse through the feedback policy K_t. Unfortunately, this feedback assumes the system is unconstrained linear-quadratic and will not allow recourse for non-convex obstacle constraints. We also suspect control saturation will make these optimistic in practice. A common heuristic for POMDPs is maximum-likelihood observations, but this underestimates uncertainty and is not suitable here. Finally, recent work include sidestepping the problem by learning sufficiently accurate prediction models, e.g. [23], but this seems like a strong assumption.

Here we instead propose a novel policy search method to efficiently compute a probabilistic constraint approximation *including* recourse. This allows any obstacle behavior model and is not tied to linear-Gaussian models. For our experiments in section VI, each human follow acceleration and velocity profiles from [19] and is simply given a sequence of target positions based on the scenario.

IV. TRAJECTORY-POLICY APPROXIMATIONS

Policy search is a stochastic optimal control approach often used for reinforcement learning problems in robotics [24], [9]. The controls are determined by a global policy function $\mathbf{u}_t = \pi_\theta(\mathbf{x}_t)$, with a parameter vector θ. This allows approximate global solutions to stochastic problems, but we can only manipulate the state and control inputs through the policy parameters θ.

In some cases this problem admits closed form solution, but in the general case θ can be learned by episodic simulations over a target scenario, either using a model or with the real system in the loop. While being stochastically correct, the problem is expensive and θ is usually limited to a small number of parameters. Since the policy function is global and needs to be defined over the entire state space $\mathbf{x} \in \mathbb{R}^n$, this means higher-dimensional state spaces require expertly structured policy representations. Policies are typically composed of simpler control laws and motor primitives [10]. Constraints are also problematic and are typically included as costs when permissible.

Trajectory optimization on the other hand, while being difficult for the general stochastic case, is otherwise typically only computationally cubic in the state space. In addition, it has a mature theory for deterministic constrained optimization.

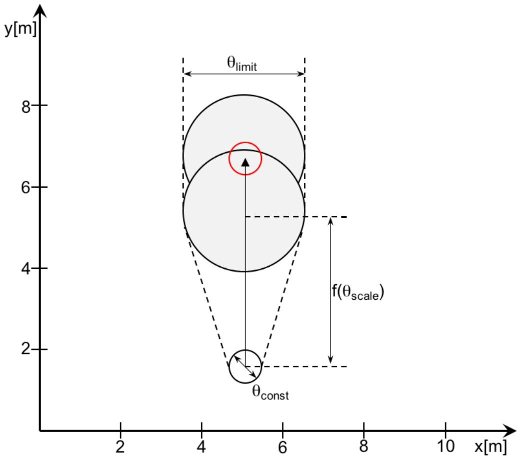


Fig. 2: Parametric safety margin.

To exploit the advantages of both we propose a novel approximation to the stochastic constrained trajectory optimization problem in Eq. (1). By using a *policy* $\pi_\theta(\mathbf{x}_t)$ represented by the output $\mathbf{u}_{t=0}$ of a fast deterministic model-predictive control solver. To the best of our knowledge this has never been attempted before. In particular, we want to approximate difficult constrained stochastic problems by manipulating deterministic soft constraints, using a suitable parametric safety margin $\mathbf{m}(\theta, \mathbf{x}_t)$. Instead of difficult probabilistic constraints of the type $\mathrm{p}(\mathbf{g}(\mathbf{x}_t, \mathbf{u}_{t-1}) \geq \mathbf{0}) > p$ we have $\mathbf{g}(\mathbf{x}_t, \mathbf{u}_{t-1}) \geq \mathbf{m}(\theta, \mathbf{x}_t)$ *inside* the *MPC policy*, where θ is the policy parameter vector. By using policy search, this can be learned to satisfy probabilistic constraints with confidence p over an *entire scenario*

$$\begin{aligned} &\underset{\theta}{\arg\min} \quad \mathbb{E}\left[\sum_{t=1}^{T_{\text{scenario}}} c(\mathbf{x}_t, \pi_\theta(\mathbf{x}_{t-1}))\right] \\ &\text{subject to} \\ &\qquad\qquad \mathrm{p}(\mathrm{dist}(\mathbf{p}_{r,t}, \mathbf{p}_{o,t}) > 0, \forall t, \forall o) > p. \end{aligned} \tag{3}$$

We defer the specifics of learning θ to section IV-A and now consider probabilistic collision constraints and deterministic approximations of the type

$$\mathbb{E}[\mathrm{dist}(\mathbf{p}_{\mathrm{r,t}}, \mathbf{p}_{\mathrm{o,t}})] > m(\theta, \mathbf{x}_t). \tag{4}$$

We start from the simple constraints on predictive distribution from Eq. (2). As discussed in sections II and III, there were two problems with these. First, since controller recourse was not adequately captured, they grew increasingly pessimistic for non-cooperative obstacles. Second, per-step confidences of the type in Eq. (1) are generally difficult to map to an overall safety level, especially for MPC. Here we address both of these by introducing *limit* and *scaling* parameters, respectively. The intuition behind this model is that at some point the controller will have adjusted to a failed prediction, which means that further predictive uncertainty can be disregarded. The scaling parameter allows us to calibrate the per-step uncertainties to satisfy a desired overall confidence level for a scenario.

The overall shape of the safety margins will therefore be spheres around each step of the trajectory, see Fig. 2. They will increase in size with a rate controlled by θ_{scale}, up until some limit θ_{limit} in meters. We also include a constant θ_{const} to capture bias such as latency.

Formally, to approximate the stochastic collision avoidance scenario we use the safety margin function $m(\theta_{\text{limit}}, \theta_{\text{scale}}, \theta_{\text{const}}, \mathbf{x}_t) = \min(\theta_{\text{scale}} \cdot n(\mathbf{x}_t) + \theta_{\text{const}},\ \theta_{\text{limit}})$. Where $n(\mathbf{x}_t) = \sqrt{\chi_2^2(1-0.95)\,\mathrm{var}_{\max}(\mathbf{p}_{r,t} - \mathbf{p}_{o,t})}$ is the original linear-Gaussian approximation from Eq. (2).

We could have used any parameterized function, but $n(\mathbf{x}_t)$ serves as a baseline for comparison. We fixed θ_{const} to $0.4\,\mathrm{m}$ to account for worst-case latencies in the system, the rest were learned by Bayesian optimization.

A. Constrained Bayesian Policy Optimization

An agile robot may only need a small safety margin, while a less nimble robot may need a bigger one. Here we focus on learning these parameters using Bayesian policy optimization on simulations of the system. Bayesian optimization has previously been used for policy optimization, most recently in [25]. Here we expand upon this by using *constrained* Bayesian optimization to solve the novel constrained policy search problem in Eq. (3).

Bayesian optimization is a recent method for global optimization that typically uses Gaussian processes to model the outcome of sampling the parameter space of a cost function.

In brief, a Gaussian process is defined as a set of random variables, any finite number of which have a joint Gaussian distribution [26]. The process is completely specified by a mean function $m(x)$ and a covariance function $k(x, x')$ that are functions of the input variables. For clarity we assume that all data is standardized with zero mean, turning the covariance function into $k(\mathbf{x}, \mathbf{x}') = \mathbb{E}[f(\mathbf{x})f(\mathbf{x}')]$. This defines the covariance between input points such that the distribution of any points on $f(\mathbf{x})$ is completely specified by a joint multivariate Gaussian. By conditioning on the known data and marginalizing out the hyperparameters in the covariance function, one can learn a distribution over functions $f(\mathbf{x})$ such that mean and confidence intervals over any new point x^* can be computed.

In Bayesian optimization, the learned GP $f(\mathbf{x})$ is used as a surrogate model for the cost function $c(\mathbf{x})$ to find the most beneficial points to sample according to some *acquisition function*. Typical examples are expected improvement, UCB and predictive entropy. Since Gaussian processes are highly data efficient on smooth surfaces, points carry considerable information on surrounding points.

Recent advances [12][13], whose work our policy search is built upon, extend this to *constrained* Bayesian optimization, where not only a cost function $c(\mathbf{x})$ is learned, but also a constraint function $g(\mathbf{x})$. Since GPs are Bayesian, they can handle probabilistic constraints that we need to solve our novel constrained policy search problem in Eq. 3.

V. Case Study: Safe Quadcopter MPC

While the proposed approach generalizes to any non-linear dynamics and constraints, fast off-the-shelf solvers [11],[14]. so far only exist for the constrained convex case. Here we focus on the special case of otherwise convex problems with concave geometric collision constraints and describe the real-time quadcopter implementation used in the experiments. The underlying deterministic MPC controller takes the standard form of Eq. (5).

$$\operatorname*{arg\,min}_{\mathbf{u}_0\ldots\mathbf{u}_{T-1},\mathbf{x}_1\ldots\mathbf{x}_T,\epsilon_{1..T}} \sum_{t=1}^{T} c(\mathbf{x}_t,\mathbf{u}_{t-1})$$

$$\begin{aligned} &\text{subject to} && (5)\\ &\mathbf{x}_t = f(\mathbf{x}_{t-1},\mathbf{u}_{t-1};\theta_{\text{dyn}}),\\ &\mathbf{g}_{\text{task}}(\mathbf{x}_t,\mathbf{u}_{t-1}) \geq -w_{\text{task}}^{-1}\epsilon_{t,\text{task}},\\ &\mathbb{E}[\text{dist}(\mathbf{p}_{\text{r,t}},\mathbf{p}_{\text{o,t}})] - m(\theta,\mathbf{x}_t) \geq w_{\text{obst}}^{-1}\epsilon_{t,\text{obst}},\\ &\epsilon_t \geq 0,\ \text{where } t = 1,\ldots,T. \end{aligned}$$

The objective $c(\mathbf{x}_t,\mathbf{u}_t)$ here includes goal distance $w_{\text{obj}}\|\mathbf{p}_{r,T}-\mathbf{p}_g\|^2$, but this can also be handled by a soft constraint. We found large control penalties $w_{\text{control}}\|\mathbf{u_t}\|^2$ to improve performance. Since the controller preferred small control inputs, it reserved recourse to deal with unpredictable obstacles. The objective also contains penalty terms for violation of elastic constraints $\|\epsilon_t\|^2$.

The dynamics $f(.)$ were approximated by a linear function and learned from data. The task constraints included hard control saturation box constraints, as well as elastic box constraints, for speed, which are all also convex. We prioritized them such that $w_{\text{obst}} \approx w_{\text{task}} > w_{\text{obj}}$ with exponential discounting over time.

The obstacle constraints from Eq. (4) are unfortunately concave. The classical approach is to use mixed-integer programming with concave polygon constraints, but this is slow and even the relaxed problem has multiple linear constraints. Using our distance functions one could do full SQP, but that is complicated and we found that just projecting the constraint directly on the geometry (sphere) worked well. Since obstacle constraints are concave, each point along the trajectory will be iteratively projected. These constraints will often conflict, but the soft constrained formulation typically converged in just a few iterations.

Local minima can be a problem when multiple obstacles are involved, but with random restarts it seems surprisingly good at avoiding bad minima. We employed a variance reduction strategy by randomizing control inputs sequentially directed into each quadrant.

Fig. 3: The LinkQuad Quadcopter.

Quadcopters are often approximated by linear models, which are conveniently convex. Using maximum likelihood we learned a simple linear model θ_{dyn} of lateral and angular dynamics from data. As is common, the on-board control system uses hierarchical PID loops to control attitude given some target control input values. For simplicity we opted not to replace those and applied MPC by considering the dynamics, including the PID loops, as part of the model. To avoid unmodeled higher order dynamics we also included control rate constraints reflecting observed control deltas in the data. Here we only control it in the plane, but 3D is likely also possible in real-time. This results in $\mathbf{x} = [x,\dot{x},\theta,y,\dot{y},\phi]$ and $\mathbf{u} = [u_\theta,u_\phi]$.

VI. Experiments

The aim of Bayesian Policy Optimization Model-Predictive Control (BPO-MPC) is to provide a holistic framework for safe collision avoidance, while still retaining real-time capability by exploiting approximations based on fast convex MPC solvers. To demonstrate the approach we attempt to set up reasonably realistic scenarios involving the quadcopter model from section V and non-cooperative moving obstacles under uncertainty from section III. We show that we can find highly accurate safety margins for different scenarios by validating the results over 12 hours of simulation. Learning the safety parameters to this degree of accuracy took hours, but more conservative solutions are found earlier. One can either use the safety parameters of one suitably conservative scenario, or select between them depending on situation. Interesting future work include learning multi-scenario safety policies on-line.

All scenarios have uncertainty in the identified dynamics model, observations, as well as in the movement of the obstacles. As outlined in section III, obstacle unpredictability is crucial for non-cooperative obstacles, which is the main focus of this work. While our framework will work with any observation model, we use a standard Kalman filter with a constant velocity model. Since we only need to measure the relative distance to the obstacle using a range finder, the sensor noise σ_v is set to a relatively low value of 1 cm. We expect actual on-board sensor uncertainty to result more from occasional artefacts rather than white Gaussian noise, but leave more accurate sensor models to future work.

The framework is implemented in Eigen/C++ and Python using a ROS distributed architecture. All experiments were run on one core of an Intel Core i7 3.4Ghz CPU. All simulated scenarios are run in soft real-time to be as realistic as possible, which includes possible artefacts from computation time and network latency that are present in the real system. Finally, we include demonstration flights with the real quadcopter platform, using real humans as obstacles. Unless noted otherwise the quadcopter speed constraint was set to $1.5\,\mathrm{m\,s^{-1}}$.

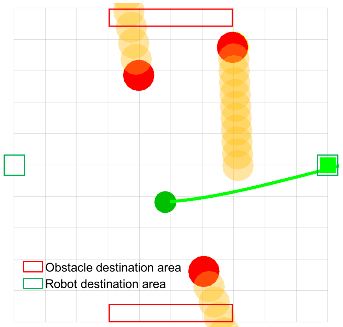


Fig. 4: The intersection scenario, crossing a street.

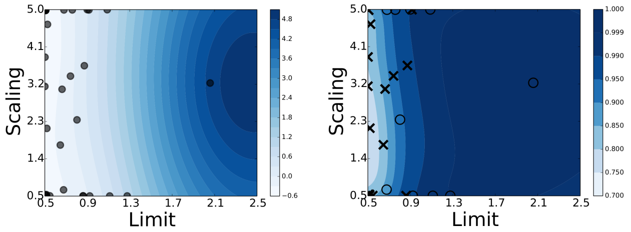
A. Simulated Intersection Scenario

With autonomous cars on the horizon and the popularity of consumer UAVs rising, their behavior in traffic is of great importance. In the first scenario we simulate the quadcopter from section V safely crossing a street.

We use the previously defined human obstacle model and explore the safety radius required for a street with three pedestrians. Each pedestrian is given a random destination at either end of the street in a 10 m long and 4 m wide area. They select a new destination either upon reaching their current one, or at random every 10-20 s. This means that the controller has to be able to cope with moving obstacles adjusting course or turning unpredictably. It bears pointing out that humans can accelerate very quickly from a state of stand-still while many robotic platforms, including the mid-size quadcopter from section V, have slower higher-order dynamics. The scenario, shown in Fig.4, is split into a sequence of runs across a street. The quadcopter tries to safely reach its goal through traffic while the goal alternates at opposite sides of the street.

We optimize the safety parameters of BPO-MPC with constrained Bayesian optimization as described in section IV-A. We take the mean time to reach the goal as the objective cost, and the safety probability *per minute* as the constraint. The resulting posterior surface approximations are seen in Fig.5a for the cost, and Fig. 5b for the estimated safety probability. For convenience the experiments are aggregated into 20 minute runs, represented as a success (circle) or failure (cross) in the figure.

In Table I we list actual outcomes of parameters learned by BPO-MPC for different safety levels and compare them with the obstacle motion models discussed in section III. These are confidence intervals based on the random acceleration model, as well as the deterministic upper bound.



(a) Expected cost (normalized). (b) Expected safety level.

Fig. 5: Bayesian policy optimization of safety parameters for the intersection scenario.

TABLE I: Results from intersection scenario for different algorithms and target safety levels. Actual safety level estimated over 12 h.

Safety constraints	θ_{limit}	θ_{scale}	Cost	Est. p
BPO-MPC $p = 0.95$	0.8 m	0.6	9.82	0.956
BPO-MPC $p = 0.99$	0.9 m	0.6	9.96	0.985
Prior assump. $p = 0.95^1$	∞	1	-	-
Determ. Bound	-	-	36.36	0.9875

As can be seen The BPO-MPC controller solves the problem with a low cost while being close to the target safety level. We note that for BPO-MPC, $p = 0.99$ is per minute of non-cooperative interaction, resulting in a cautious yet efficient controller. The other motion models are extremely pessimistic. The random acceleration model resulted in such wide safety margins that it was unable to reach the goal even at a $p = 0.95$ level. The deterministic bound was also very conservative and completed the task only with high cost. We also noted some degradation in MPC performance as the large margins resulted in a need for longer trajectories. While the confidence level of the random acceleration model could be adjusted by hand, unlike the proposed approach, it is unclear how to relate that to more meaningful units like minutes of scenario time. As discussed in section IV, since they do not take the effect of the controller recourse into account, the overall shape of their safety region will also be less useful for non-cooperative obstacles.

B. Simulated Warehouse Scenario

Safe interaction with humans is key to continued automation and acceptance of robots into workplaces. The warehouse scenario was made to imitate the conditions a robot might face working alongside humans in a warehouse or assembly facility. Recently, commercial entities like Amazon have expressed interest in using quadcopters for delivery. In the warehouse scenario pictured in Fig. 6 a quadcopter is given targets shown as green boxes, while avoiding human workers freely roaming around the room. These could, for example, be packages to pick up or areas to inspect. New targets are continuously spawned in 15s intervals.

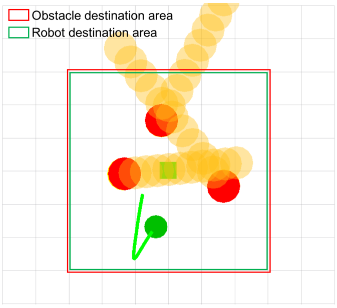


Fig. 6: The warehouse scenario.

We again give humans right-of-way by using the non-cooperate assumption. Humans are given random destinations in a 6x6m square, again using the model from section III, but giving them new destinations with shorter, uniformly distributed, intervals of 6 to 10s. This forces the controller into intense avoidance situations as it tries to stay in-between randomly moving obstacles to reach its targets[2].

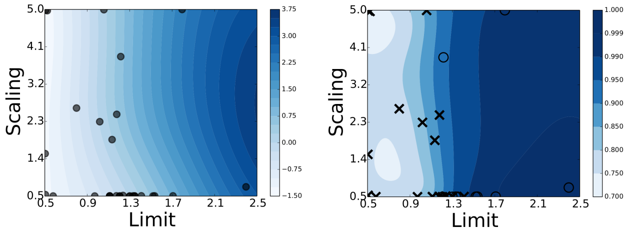
TABLE II: Warehouse scenario for different target safety levels. Actual safety level estimated over 12 h.

Safety constraints	θ_{limit}	θ_{scale}	Cost	Est. p
BPO-MPC $p = 0.95$	1.3 m	0.5	2.66	0.954
BPO-MPC $p = 0.99$	1.5 m	0.6	2.82	0.985
BPO-MPC $p = 0.999$	1.8 m	0.6	3.13	0.997
BPO-MPC $p = 0.999+$	2.1 m	1.0	3.84	1.00

As can be seen in Table II the BPO-MPC controller can safely navigate in this scenario. The Gaussian process safety approximation gets a bit optimistic for the higher safety levels, likely due to the actual distribution having thicker tails. If more exact probabilities are needed it still provides a good starting point for manual tuning over longer and more expensive simulation runs. To illustrate that the probability of collision tends to zero we also significantly perturbed the safety parameters towards even safer regions in Fig. 7b, shown as $p = 0.999+$ in the table.

We also examined what effect the number of obstacles has. We use a safety level of $p = 0.99$ per minute. The results for the scenario with one to three humans can be seen in Table III. Most prominently, we see that the safety margins steeply increase already when going from one to two humans. As noted in section III this is not unexpected since constraint saturation should limit the recourse available to dodge a secondary obstacle when already dodging one.

With even more obstacles, all escape routes can easily be blocked and ultimately box the robot in. Due to the physical constraints of the robot, this becomes an increasingly ill-posed problem, and we expect that this will lead to very conservative safety margins for large and dense crowds. When given no recourse, humans can call attention to themselves to trigger cooperative behavior, like using a horn on a car. We suspect a similar backup behavior could be employed for robots, but we leave navigating such semi-cooperative crowds to future work.



(a) Expected cost (normalized). (b) Expected safety level.

Fig. 7: Bayesian policy optimization of safety parameters for the warehouse scenario.

TABLE III: Optimization results from warehouse scenario with different numbers of humans

Safety constraints	Humans	θ_{limit}	θ_{scale}	Cost
BPO-MPC $p = 0.99$	1	0.8 m	0.6	1.35
BPO-MPC $p = 0.99$	2	1.1 m	0.6	2.32
BPO-MPC $p = 0.99$	3	1.5 m	0.6	2.82

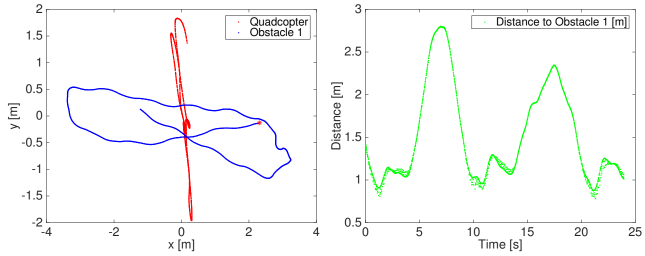
C. Real Quadcopter Flight Scenarios

We flew two scenarios with the real LinkQuad quadcopter from Fig.3. For convenience the quadcopter was controlled over a serial link from a ground station, although it is currently possible for a platform of this size to carry an Intel NUC with comparable performance on-board.

For in-door positioning we used a VICON motion capture system and added sensory noise to the obstacle distance corresponding to the previous assumptions. Limited size of the tracking area introduced the additional requirement of a safety margin small enough for meaningful interaction. We therefore lowered the controller speed constraint to $1\,\mathrm{m\,s^{-1}}$, as well as obstacle speed to a casual walk. As seen in Table III, simultaneous interaction with multiple non-cooperative obstacles can also have a large impact on the required safety margin. To minimize this, obstacles moved in non-overlapping segments. We found safety parameters for this configuration and added an 0.2m extra for any unmodelled effects like jitter in the serial link, rounding the safety parameters up to 1.0m limit and 0.6m scaling.

We show two examples from multiple real flights in Fig. 8 and 9. In the first scenario one person is repeatedly attempting to walk through the space occupied by the quadcopter. While a reactive controller would have bounced away, by planning for the movement it smoothly moves to the side and back, being the safe trajectory with lowest cost. In the second scenario the quadcopter is trying to move diagonally through the area while two humans walk around randomly. At first it has to reverse to go around the red obstacle, passing it on the right, and then swerving to the left to dodge the blue obstacle coming in from the right.

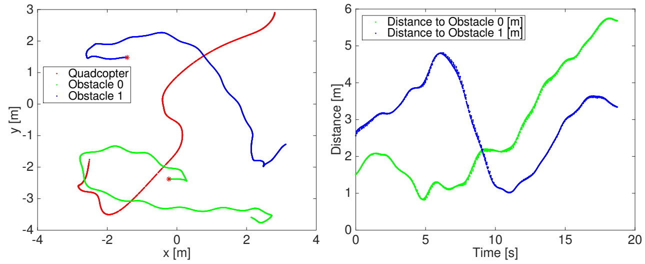
[2] See supplementary video material.



(a) Trajectories.

(b) Distance to obstacle.

Fig. 8: Quadcopter avoiding one human obstacle.



(a) Trajectories.

(b) Distance to obstacles.

Fig. 9: Quadcopter avoiding two human obstacles.

Although a small sample, as can be seen from the distance plots, the chosen 1 m safety margin was never close to being exhausted.

VII. CONCLUSIONS

We examined the problem of autonomous robots safely navigating environments populated by humans, without prior coordination. This was reduced to stochastic trajectory optimization around non-cooperative obstacles. Common stochastic simplifications lead to poor results and we instead derived a novel stochastic approximation by combining Bayesian policy optimization with fast solvers from model-predictive control. This enabled us to construct holistic controllers that can solve the collision avoidance problem for a desired level of confidence while taking both uncertainty and dynamics into account. We found the proposed approach accurate on simulated scenarios and demonstrated a real-time implementation by flying a real quadcopter. It produced accurate control with safety margins tight enough for use on real robots.

REFERENCES

[1] M. Hoy, A. S. Matveev, and A. V. Savkin, "Algorithms for collision-free navigation of mobile robots in complex cluttered environments: a survey," *Robotica*, vol. 33, pp. 463–497, 3 2015.

[2] P. Fiorini and Z. Shiller, "Motion planning in dynamic environments using velocity obstacles," *The International Journal of Robotics Research*, vol. 17, no. 7, pp. 760–772, 1998.

[3] C. Fulgenzi, A. Spalanzani, and C. Laugier, "Dynamic obstacle avoidance in uncertain environment combining pvos and occupancy grid," in *Robotics and Automation (ICRA), 2007 IEEE International Conference on*. IEEE, 2007, pp. 1610–1616.

[4] J. Van den Berg, D. Wilkie, S. J. Guy, M. Niethammer, and D. Manocha, "Lqg-obstacles: Feedback control with collision avoidance for mobile robots with motion and sensing uncertainty," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*. IEEE, 2012, pp. 346–353.

[5] J. Snape, J. van den Berg, S. Guy, and D. Manocha, "The hybrid reciprocal velocity obstacle," *Robotics, IEEE Transactions on*, vol. 27, no. 4, pp. 696–706, Aug 2011.

[6] D. Bareiss and J. van den Berg, "Reciprocal collision avoidance for robots with linear dynamics using lqr-obstacles," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, May 2013, pp. 3847–3853.

[7] J. van den Berg, S. Patil, and R. Alterovitz, "Motion planning under uncertainty using iterative local optimization in belief space," *The International Journal of Robotics Research*, vol. 31, no. 11, pp. 1263–1278, 2012.

[8] M. P. Vitus and C. Tomlin, "Closed-loop belief space planning for linear, gaussian systems," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, May 2011, pp. 2152–2159.

[9] M. P. Deisenroth, G. Neumann, J. Peters, *et al.*, "A survey on policy search for robotics." *Foundations and Trends in Robotics*, vol. 2, no. 1-2, pp. 1–142, 2013.

[10] J. Kober and J. Peters, "Policy search for motor primitives in robotics," *Mach. Learn.*, vol. 84, no. 1-2, pp. 171–203, July 2011.

[11] A. Domahidi, A. Zgraggen, M. Zeilinger, M. Morari, and C. Jones, "Efficient interior point methods for multistage problems arising in receding horizon control," in *IEEE Conference on Decision and Control (CDC)*, Maui, HI, USA, Dec. 2012, pp. 668 – 674.

[12] M. Gelbart, J. Snoek, and R. Adams, "Bayesian optimization with unknown constraints," in *Proceedings of the Thirtieth Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-14)*. Corvallis, Oregon: AUAI Press, 2014, pp. 250–259.

[13] J. M. Hernández-Lobato, M. A. Gelbart, M. W. Hoffman, R. P. Adams, and Z. Ghahramani, "Predictive entropy search for bayesian optimization with unknown constraints," in *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, 2015, pp. 1699–1707.

[14] H. J. Ferreau, C. Kirches, A. Potschka, H. G. Bock, and M. Diehl, "qpoases: A parametric active-set algorithm for quadratic programming," *Mathematical Programming Computation*, pp. 1–37, 2013.

[15] L. Blackmore and M. Ono, "Convex chance constrained predictive control without sampling," in *Proceedings of the AIAA Guidance, Navigation and Control Conference*, 2009, pp. 7–21.

[16] J. Nocedal and S. J. Wright, *Numerical Optimization*, 2nd ed. New York: Springer, 2006.

[17] M. Ono and B. Williams, "Iterative risk allocation: A new approach to robust model predictive control with a joint chance constraint," in *Decision and Control (CDC). 47th IEEE Conference on*, Dec 2008, pp. 3427–3432.

[18] D. Althoff, M. Althoff, D. Wollherr, and M. Buss, "Probabilistic collision state checker for crowded environments," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, May 2010, pp. 1492–1498.

[19] M. Moussaïd, D. Helbing, S. Garnier, A. Johansson, M. Combe, and G. Theraulaz, "Experimental study of the behavioural mechanisms underlying self-organization in human crowds," *Proceedings of the Royal Society B: Biological Sciences*, vol. 276, no. 1668, pp. 2755–2762, 2009.

[20] J. van den Berg and M. Overmars, "Planning time-minimal safe paths amidst unpredictably moving obstacles," *The International Journal of Robotics Research*, vol. 27, no. 11-12, pp. 1274–1294, 2008.

[21] D. P. Bertsekas, *Dynamic Programming and Optimal Control*, 2nd ed. Athena Scientific, 2000.

[22] D. Mayne, "A second-order gradient method for determining optimal trajectories of non-linear discrete-time systems," *International Journal of Control*, vol. 3, no. 1, pp. 85–95, 1966.

[23] G. S. Aoude, B. D. Luders, J. M. Joseph, N. Roy, and J. P. How, "Probabilistically safe motion planning to avoid dynamic obstacles with uncertain motion patterns," *Autonomous Robots*, vol. 35, no. 1, pp. 51–76, 2013.

[24] J. Peters and S. Schaal, "Policy gradient methods for robotics," in *Intelligent Robots and Systems (IROS), 2006 IEEE/RSJ International Conference on*. IEEE, 2006, pp. 2219–2225.

[25] A. Cully, J. Clune, D. Tarapore, and J.-B. Mouret, "Robots that can adapt like animals," *Nature*, vol. 521, no. 7553, pp. 503–507, 2015.

[26] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. Cambridge, MA: MIT Press, 2006.