Moving Obstacle Avoidance: a Data-Driven Risk-Aware Approach

Skylar X. Wei*, Anushri Dixit*, Shashank Tomar, and Joel W. Burdick, *Member, IEEE*

***Abstract*—This paper proposes a new structured method for a moving agent to predict the paths of dynamically moving obstacles and avoid them using a risk-aware model predictive control (MPC) scheme. Given noisy measurements of the a priori unknown obstacle trajectory, a bootstrapping technique predicts a set of obstacle trajectories. The bootstrapped predictions are incorporated in the MPC optimization using a risk-aware methodology so as to provide probabilistic guarantees on obstacle avoidance. We validate our methods using simulations of a multi-rotor drone that avoids various moving obstacles.**

***Index Terms*—Predictive control for linear systems, Stochastic optimal control, Uncertain systems, Robotics.**

I. INTRODUCTION

EMERGING applications of robots in urban, cluttered, and potentially hostile environments have increased the importance of online path planning with obstacle behavior classification and avoidance [1]. Traditionally, robot-obstacle interaction is formulated as the problem of planning a collision-free path from a starting position to a goal [2]. In environments with an arbitrary number of moving obstacles and agents with bounded velocity, this problem is known to be NP-hard [3].

One way to handle dynamic obstacles is to limit their modeled motions. In [4], the authors assumed a priori knowledge of the obstacle dynamics or motion patterns. Or, one can plan the agent's path off-line using a Probabilistic Roadmap (PRM) in a field of static obstacles and then replan when dynamical behaviors are observed [5]. However, without prior knowledge of an obstacle's behavior, a worst-case analysis of unsafe set can lead to conservative behavior. Potential fields (PFs) are actively used for dynamic obstacle avoidance: e.g., Lam et al. [6] apply artificial PFs with stochastic reachable sets in Human-Centered environments. Slow moving and simple (linear or double integrator-like) dynamics are assumed. Switching-based planning methods detect and classify dynamic obstacle behavior against a set of trajectories, such as constant speed, linear, and projectile-like motion [7], [8]. Classification-based methods require distinguishable obstacle behaviors and prior knowledge about the dynamic environment to generate set trajectories.

This paper presents a new framework for discovering the dynamics of a priori unknown moving obstacles, forecasting their trajectories, and providing risk-aware optimal avoidance strategies. It replaces the need for obstacle trajectory/model classification while allowing online computation. Extracting a dynamics model from data is challenging [9], especially when the available data is limited, noisy, and partial. To tackle partial measurements, we leverage Takens' embedding theorem [10], which uses partial observations to produce an attractor that is diffeomorphic to the full-state attractor. We then use Singular Spectrum Analysis (SSA) [11], [12] to separate noise from the underlying signal and to extract a predictive model of obstacle behavior. Our use of time delay embedding is also the basis of Eigensystem Realization Algorithm (ERA) in linear system identification [13]. Inspired by [14], we use a classical bootstrap to forecast a set of obstacle trajectories with statistical quantification. An MPC planner then incorporates the set of obstacle forecasts as an affine conservative approximation of a distributionally-robust chance constraint (DRCC). This constraint is then efficiently recast in a risk-aware manner, allowing an MPC optimization based on sequential convex programming [15], [16].

We demonstrate our approach on three scenarios that exhibit increasingly complicated dynamical behavior. Monte-Carlo simulations verify the planner's ability to uphold the user chosen chance constraint. The risk-aware reformulation not only gives provable probabilistic collision avoidance guarantees, but also allows an on-line execution of the planner.

Notation: The set of positive integers, natural numbers, real numbers, and positive reals are denoted as $\mathbb{Z}_+$, $\mathbb{N}$, $\mathbb{R}$, and $\mathbb{R}_+$, respectively. We denote the sequence of consecutive integers $\{i, \cdots, i+k\}$ as $\mathbb{Z}^{i:i+k}$. The finite sequence $\{a_1, \cdots, a_k\}$ of scalars or vectors a is denoted as $\{a\}_1^k$. The expression $I_{n\times n}$ denotes n by n identity matrices and $\mathbf{1} = [1, 1, 1]^T$.

II. SSA PRELIMINARIES

Consider a discrete-time multivariate stochastic process $\{\boldsymbol{o}^m\}_1^N$ where m denotes the m^{th} observable of the process out of the total N^o available observables, and N is the number of available observations. Suppose that the true stochastic process model of the observables is $\hat{\boldsymbol{o}}^m = \boldsymbol{o}^m + \boldsymbol{\gamma}^m$, where $\boldsymbol{\gamma}^m$ denotes a random discrete-time zero-mean measurement noise, and $\boldsymbol{o}^m$ is the noiseless observable that captures the governing laws, which can be composed of *trends*, *seasons*, and *stationary* time series. Singular Spectrum Analysis [11] separates the true signal $\boldsymbol{o}^m$ and the noise $\boldsymbol{\gamma}^m$ and extracts a recursive governing dynamic model of $\boldsymbol{o}^m$ that can generate a short term accurate forecast. Fig. 1 describes this method.

1) Time Delay Embedding: Takens' method of delays [10] can reconstruct qualitative features of the full-state phase-space from delayed partial observations. The m^{th}-state observables $\hat{\boldsymbol{o}}^m$ are delay embedded into a trajectory (Hankel) matrix $H^m_{[L\times N]}$, Fig. 1 gives an example of the Hankel matrix for state x. Parameter L is the time delay length, and N is the

*Both authors contributed equally.

The authors are with the Division of Engineering & Applied Science, California Institute of Technology, MC 104-44, Pasadena, CA 91125, ({swei, adixit, stomar, jburdick}@caltech.edu).

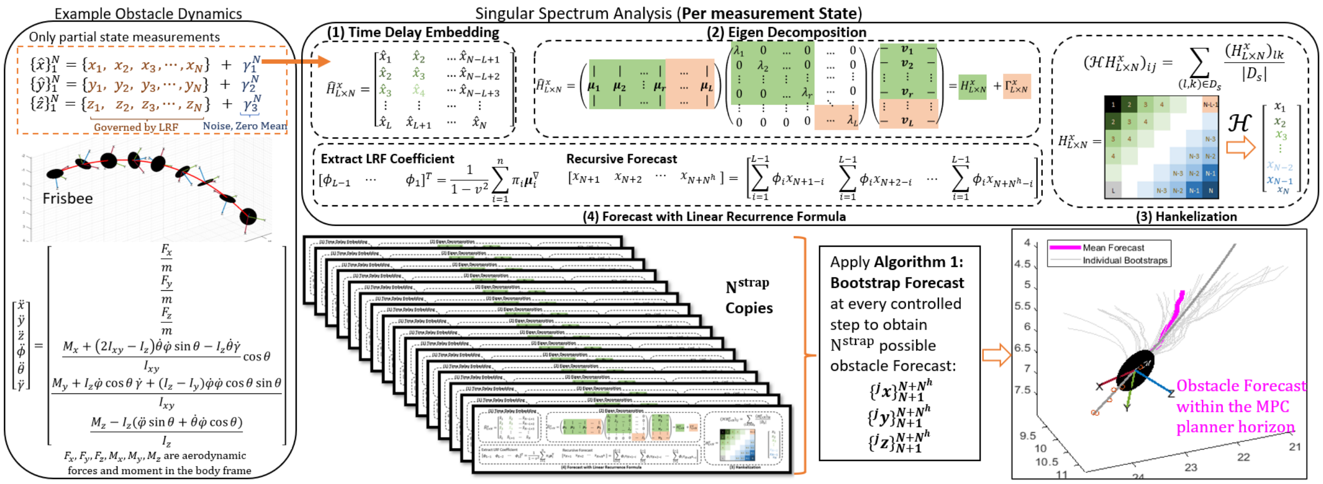
Example Obstacle Dynamics
Only partial state measurements
$\{\hat{x}\}_1^N = \{x_1, x_2, x_3, \cdots, x_N\} + \gamma_1^N$
$\{\hat{y}\}_1^N = \{y_1, y_2, y_3, \cdots, y_N\} + \gamma_2^N$
$\{\hat{z}\}_1^N = \{z_1, z_2, z_3, \cdots, z_N\} + \gamma_3^N$
Governed by LRF
Noise, Zero Mean
Frisbee
$F_x, F_y, F_z, M_x, M_y, M_z$ are aerodynamic forces and moment in the body frame
Singular Spectrum Analysis (Per measurement State)
(1) Time Delay Embedding
(2) Eigen Decomposition
(3) Hankelization
(4) Forecast with Linear Recurrence Formula
Extract LRF Coefficient
Recursive Forecast
N^{strap} Copies
Apply Algorithm 1: Bootstrap Forecast at every controlled step to obtain N^{strap} possible obstacle Forecast:
$\{^j\boldsymbol{x}\}_{N+1}^{N+N^h}$
$\{^j\boldsymbol{y}\}_{N+1}^{N+N^h}$
$\{^j\boldsymbol{z}\}_{N+1}^{N+N^h}$
Mean Forecast
Individual Bootstraps
Obstacle Forecast within the MPC planner horizon

Fig. 1: A description of bootstrap-SSA-forecast architecture in forecasting the trajectory of a Frisbee where the stochastic observables (corrupted by zero-mean, noise) consist of $\{\hat{\boldsymbol{o}}\}_1^N = [\{\hat{x}\}_1^N, \{\hat{y}\}_1^N, \{\hat{z}\}_1^N]$, the Frisbee's center positions with respect to an inertial frame. The SSA analysis and bootstrap forecast is applied to every observable state. Despite its 12-state governing dynamics [17] and with only center position measurements of the Frisbee, we show an example N^{strap} forecasts of the Frisbee trajectory for future time steps $\{1, 2, \cdots, N^h\}$ using our proposed framework.

time series length. Repeating patterns in the Hankel matrix represent underlying trends and oscillations, which can be extracted from its covariance matrix: $X^m = H^m_{[L,N]}(H^m_{[L,N]})^T$.

2) *Eigen Decomposition:* To recover the true signal $\boldsymbol{o}^m$, we seek the best, low-rank matrix approximation of this signal by thresholding the eigenvalues of X^m [18]. The symmetric covariance matrix X^m has a spectral decomposition $U\Sigma U^T$, where Σ is a diagonal matrix with real eigenvalues $\lambda_1 \geq \lambda_2 \geq \cdots \lambda_L$. The matrix of left eigenvectors $U = [\boldsymbol{\mu}_1, \cdots, \boldsymbol{\mu}_L]$ is orthogonal. The truncated right eigenvectors $V = [\boldsymbol{\nu}_1, \cdots, \boldsymbol{\nu}_L]^T \in \mathbb{R}^{L\times N}$ of X^m can be found as $V = U\Sigma$. Suppose $\boldsymbol{\lambda}$ is the optimal threshold and $\lambda_n \geq \boldsymbol{\lambda} \geq \lambda_{n+1}$, which partitions the Hankel matrix $H^m_{[L,N]}$ as:

$$H^m_{[L,N]} = \underbrace{\sum_{p=1}^{n} \sqrt{\lambda_p}\boldsymbol{\mu}_p\boldsymbol{\nu}_p^T}_{\triangleq H^o_{[L,N]}} + \underbrace{\sum_{p=n+1}^{L} \sqrt{\lambda_p}\boldsymbol{\mu}_p\boldsymbol{\nu}_p^T}_{\triangleq H^\gamma_{[L,N]}} . \tag{1}$$

3) *Hankelization:* Matrix $H^o_{[L,N]}$in (1) should maintain a Hankel structure: minor variations in its k^{th} secondary diagonals result from insufficient noise removal.[1] A *Hankelization step* performs secondary diagonal averaging in order to find the matrix $\mathcal{H}O$ that is closest to $H^o_{[L,K]}$ with respect to the Frobenius norm among all $L \times N$ Hankel matrices [11]. The operator $\mathcal{H}$ acting on $L \times N$ matrix $H^y_{[L,N]}$ entry wise is defined as: for the $(i,j)^{th}$ element of matrix $H^o_{[L,N]}$ and $i+j = s$, define a set $D_s \triangleq \{(l,n) : l+n = s, l \in \mathbb{Z}^{1:L}, n \in \mathbb{Z}^{1:N}\}$, is mapped to $(i,j)^{\text{th}}$ element of the Hankelized $\mathcal{H}H^o_{[L,N]}$ via the expression in Fig.1 (for the case of $\boldsymbol{o}^m = \boldsymbol{x}$), where $|D_s|$ denotes the number of elements in set D_s .

4) *Forecast with Linear Recurrence Formula:*

Definition 1. *A time series $Y_N = \{y\}_1^N$ admits an L-decomposition of order not larger than d, denoted by $\text{ord}_L(Y_N) \leq d$, if there exist two systems of functions $\varrho_k : \mathbb{Z}^{0:L-1} \to \mathbb{R}, \vartheta_k : \mathbb{Z}^{0:L-1} \to \mathbb{R}$, such that $y_{i+j} = \sum_{k=1}^d \vartheta_k(i)\varrho_k(j)$, $\{i,j\} \in \mathbb{Z}^{0:L-1} \times \mathbb{Z}^{0:L-1}$ $\forall k \in \mathbb{Z}^{1:d}$.* If $\text{ord}_L(Y_N) = d$, then the series Y_N admits a *L-decomposition of the order d* and both systems of functions $(\varrho_1, \cdots, \varrho_d)$ and $(\vartheta_1, \cdots, \vartheta_d)$ are linearly independent [19].

Definition 2. *A time series $\{y\}_1^N$ is governed by a linear recurrent relations/formula (LRF), if there exist coefficients $\{\phi\}_1^d$ and $\phi_d \neq 0$ such that*

$$y_{i+d} = \sum_{k=1}^{d} \phi_k y_{i+d-k}, \quad \forall i \in \mathbb{Z}^{0:N-d}, d < N\,. \tag{2}$$

Real-valued time series governed by LRFs consists of sums of products of polynomials, exponentials and sinusoids [11].

Theorem 1. *[11] Let $\boldsymbol{\mu}_i^{1:L-1}$ be the vector of the first $L-1$ components of a left eigenvector $\boldsymbol{\mu}_i$ of $H^m_{[L,N]}$, and let π_i be the L^{th} component of eigenvector $\boldsymbol{\mu}_i$. Let $v^2 \triangleq \sum_{i=1}^d \pi_i^2$. Under Assumptions 2 and 3 (see below), the LRF coefficients ϕ_i where $i \in [1, L-1]$ can be computed as:*

$$\begin{bmatrix}\phi_{L-1} & \phi_{L-2} & \cdots & \phi_1\end{bmatrix}^T = \frac{1}{1-v^2}\sum_{i=1}^{d} \pi_i \boldsymbol{\mu}_i^{1:L-1}, \tag{3}$$

and $\boldsymbol{y}$ evolves as the LRF: $\boldsymbol{y}_{N+1} = \sum_{j=1}^{L-1} \phi_j \boldsymbol{y}_{N-j}$.

III. Problem Statement

Consider the linear, discrete-time dynamical agent model:

$$\boldsymbol{x}_{i+1} = A\boldsymbol{x}_i + B\boldsymbol{u}_i, \qquad \boldsymbol{y}_{i+1} = G\boldsymbol{x}_{i+1}, \tag{4}$$

where $\boldsymbol{x}_i \in \mathbb{R}^{n_{\boldsymbol{x}}}$, $\boldsymbol{u}_i \in \mathbb{R}^{n_{\boldsymbol{u}}}$, and $\boldsymbol{y}_i \in \mathbb{R}^{n_{\boldsymbol{y}}}$, for all $i \in \mathbb{N}$, correspond to the system states, controls, and outputs at time index i respectively. The state transition, actuation, and measurement matrices are $A \in \mathbb{R}^{n_{\boldsymbol{x}} \times n_{\boldsymbol{x}}}$, $B \in \mathbb{R}^{n_{\boldsymbol{x}} \times n_{\boldsymbol{u}}}$, and $G \in \mathbb{R}^{n_{\boldsymbol{y}} \times n_{\boldsymbol{x}}}$ respectively. Constant matrix $C \in \mathbb{R}^{3\times n_{\boldsymbol{x}}}$ maps the system's states (4) to the system's x, y, z positions with respect to inertial frame E. We model the k^{th} obstacle, $k \in \mathbb{Z}^{1:N^{\text{obs}}}$, as a sphere. The obstacle occupies the point set $\mathcal{O}_k(\boldsymbol{c}_k, r_k) = \{\boldsymbol{x} \in \mathbb{R}^3 : \|\boldsymbol{c}_k - \boldsymbol{x}\|_2 \leq r_k\}$, where $\boldsymbol{c}_k \in \mathbb{R}^3$ and $r_k \in \mathbb{R}_+$ are the k^{th} obstacle's center and radius.

We consider the case where the agent (4) is tasked with following a reference output trajectory $\boldsymbol{y}^{\text{ref}}$ which need not consider obstacle information. While following this path, the

[1] The k^{th} secondary diagonals of a matrix M are also the k^{th} diagonals of M flipped horizontally with respect to its middle column.

agent may encounter N^{obs} spherical, stationary or moving obstacles. The obstacle-free region is the open set:

$$\mathcal{S} \triangleq \left\{ \mathbb{R}^3 \setminus \cup_{k=1}^{N^{\text{obs}}} \mathcal{O}_k \right\}. \tag{5}$$

Assumption 1. *Obstacles can be detected and localized at the same rate (f^+ Hz) as the planner update. Only measurements of an obstacle's geometric center with respect to frame E are assumed, and they are corrupted by a zero-mean noise. We can estimate the radius, r_k, of the k^{th} obstacle as $\hat{r}_k$, and the estimate satisfies $\hat{r}_k \geq r_k$.* [2]

Assumption 2. *All obstacle measurements, admit an L-decomposition of order d, are governed by LRFs* (2) *whose LRF coefficients can be uniquely defined.*

Assumption 3. *We assume that the obstacles' velocities are bounded by v_{max}, and the initial distances between all obstacles and the agent are significantly greater than $\frac{dv_{max}}{f^+}$.*

Problem 1. ***[Prediction]*** *Consider a multivariate stochastic process where observables $\{x\}_1^N$, $\{y\}_1^N$, and $\{z\}_1^N$ denote the spherical obstacle's true center location in reference frame, E. The measurements are corrupted by independent, zero-mean noises $\{\gamma_1\}_1^N$, $\{\gamma_2\}_1^N$, and $\{\gamma_3\}_1^N$ (see Fig. 1). Under Assumptions 1-3, we seek to predict the obstacle position at times $N+1$ to $N+N^h$ using these measurements.*

Due to limited and noisy partial data and the lack of explicit dynamics models, we estimate a Bootstrap distribution of the obstacle predictions, denoted by the random set $\mathcal{O}^{\text{pred}}$, from time index $N+1$ to $N+N^h$ and calculate its first and second moments. We account for errors in the forecastS due to poor signal and noise separation and bandwidth limits (due to limited training data and incorrect choices of embedding length L) by solving a DRCC MPC problem.

Problem 2. ***[Planning]*** *Consider the system* (4) *and free-space* (5)*. Given a discrete-time reference trajectory $\boldsymbol{y}_i^{ref} \forall i \in \mathbb{Z}^{1:N^h}$ where $N^h \in \mathbb{Z}_+$ is the length of the horizon, convex state constraints $\mathcal{D}^{\boldsymbol{x}} \subset \mathbb{R}^{n_{\boldsymbol{x}}}$, convex input constraints $\mathcal{D}^{\boldsymbol{u}} \subset \mathbb{R}^{n_{\boldsymbol{u}}}$, and a convex stage cost $L_i : \mathbb{R}^{n_{\boldsymbol{x}}} \times \mathbb{R}^{n_{\boldsymbol{u}}} \to \mathbb{R}_{\geq 0}$, a total of N^{obs} spherical obstacles each approximated by a set $\mathcal{O}_k^{pred}$, and risk tolerance $\epsilon \in (0, 1]$, we seek to compute a receding horizon controller $\{\boldsymbol{u}^*\}_1^{N_h}$ that avoids the unsafe set $\mathcal{O}^{pred} \triangleq \bigcup_{k=1}^{N^{obs}} \mathcal{O}_k^{pred}$ via the following non-convex optimization:*

$$\{\boldsymbol{u}^*\}_1^{N^h} = \min_{\{\boldsymbol{u}_k\}_1^{N^h} \in \mathbb{R}^{n_{\boldsymbol{u}}}} \sum_{i=1}^{N^h} L_i(\boldsymbol{y}_i^{ref} - \boldsymbol{y}_i, \boldsymbol{u}_i) \tag{6a}$$

$$s.t. \quad \boldsymbol{x}_{i+1} = A\boldsymbol{x}_i + B\boldsymbol{u}_i \quad \boldsymbol{y}_{i+1} = G\boldsymbol{x}_{i+1} \tag{6b}$$

$$\boldsymbol{x}_i \in \mathcal{D}^{\boldsymbol{x}}, \quad \boldsymbol{u}_i \in \mathcal{D}^{\boldsymbol{u}}, \quad \boldsymbol{x}_1 = \boldsymbol{x}_{init} \tag{6c}$$

$$\mathbb{P}(\boldsymbol{x}_i \in \mathcal{O}^{pred}) \leq \epsilon, \quad \forall i \in \mathbb{Z}^{1:N^h} \tag{6d}$$

IV. Bootstrap Forecasting

Despite empirical successes in reconstructing and forecasting [12], the theoretical accuracy of SSA is strenuous to obtain, see [20]. Inspired by [14], we use bootstrapping to improve model discovery and to produce probabilistic forecasts. Our real-time bootstrap forecast, Algorithm 1, assumes time series measurements corrupted by noise. The user-defined parameters N^{train} and N^{step} represent the number of initial training samples, and the number of newly accumulated samples during an initial bootstrap. Further, one must choose parameters δ_t and N^σ, where threshold δ_t is used to separate signal from noise, and N^σ is the number of steps of progressive relaxation of threshold δ_t .[3] In the desired signal/noise separation (1), the unknown theoretical optimal threshold $\boldsymbol{\lambda}$ must be estimated. Let $Y_N^{\lambda_1:\lambda_d}$ be the Hankelization reconstructed $\hat{\boldsymbol{y}}$ with the eigenvalues $\{\lambda\}_1^d$ and their corresponding right and left eigenvectors. Note, if $d > n$ where $\lambda_n \leq \boldsymbol{\lambda} \leq \lambda_{n+1}$, then the norm values $\|Y_N^{\lambda_1:\lambda_{d+t}} - Y_N^{\lambda_1:\lambda_{d+t+1}}\|_2 \approx \|Y_N^{\lambda_1:\lambda_{d+t+1}} - Y_N^{\lambda_1:\lambda_{d+t+2}}\|_2$ since they are comprised of the residual measurement noise. We threshold the difference between two consecutive reconstructions with δ_t/N, i.e. finding the smallest $t \in \mathbb{Z}_+$ s.t.:

$$\|Y_N^{\lambda_1:\lambda_t} - Y_N^{\lambda_1:\lambda_{t+1}}\|_2 - \|Y_N^{\lambda_1:\lambda_{t+1}} - Y_N^{\lambda_1:\lambda_{t+2}}\|_2 \leq \frac{\delta_t}{N} \tag{7}$$

Since the selection of the threshold δ_t is crucial, we add an additional parameter N^σ to ensure no principle components are lost in $Y_N^{\lambda_1:\lambda_d}$ because of bad choice of δ_t, i.e. to avoid $d < n$. We also include the next N^σ largest eigenvalues after the first t eigenvalues in the bootstrapping process. Most importantly, the number of bootstraps, N^{strap}, needs to be determined *a priori*, considering the computation capacity, number of obstacles, and the expected noise level[4].

Algorithm 1 Bootstrap Forecast Algorithms (Per Obstacle)

```
Data: Obstacle center position measurements $\{\hat{\boldsymbol{x}}\}_1^N, \{\hat{\boldsymbol{y}}\}_1^N, \{\hat{\boldsymbol{z}}\}_1^N$,
      User defined constants: $N^{\text{train}}$,$N^{\text{step}}$, $\delta_t$, $N^\sigma$, $N^{\text{strap}}$
Result: Forecast:$\{^j\boldsymbol{x}\}_{N+1}^{N+N^h}$,$\{^j\boldsymbol{y}\}_{N+1}^{N+N^h}$,$\{^j\boldsymbol{z}\}_{N+1}^{N+N^h}$,$\forall j \in \mathbb{Z}^{1:N^{\text{straps}}}$
Use $\{\hat{\boldsymbol{x}}_{N+1}, \hat{\boldsymbol{y}}_{N+1},\hat{\boldsymbol{z}}_{N+1}\}$ to update Hankel matrix
 while $istrap \leq N^{strap}$ do
    while $N+1 \geq N^{train}$ do
       for $observables = 1 : N^o$ do
          while (7) holds do
             $t_{++}$
          end
          obtain $(\{\lambda^{\text{istrap}}\}_1^t, \{\boldsymbol{\mu}^{\text{istrap}}\}_1^t, \boldsymbol{\phi}^{\text{istrap}})$ for each states, istrap$_{++}$
          for $tt = t+1 : t+N^\sigma$ do
             obtain $(\{\lambda^{\text{istrap}}\}_1^{tt}, \{\boldsymbol{\mu}^{\text{istrap}}\}_1^{tt}, \boldsymbol{\phi}^{\text{istrap}})$ for each states,
               istrap$_{++}$
          end
       end
       $N^{\text{train}} = N^{\text{train}} + N^{\text{step}}$
    end
    Back-up Strategy
end
Apply the tuples $(\{^j\lambda^{\text{istrap}}\}_1^{t_j}, \{^j\boldsymbol{\mu}^{\text{istrap}}\}_1^{t_j}, {}^j\boldsymbol{\phi}^{\text{istrap}})$ $\forall j \in \mathbb{Z}_{1:\text{Nstraps}}$ for
$x, y, z$ to the updated Hankel, where $t_j$ denotes number of eigenvalues post
truncation for the $j^{th}$ bootstrap. Perform a $N^h$ step forecast using ${}^j\boldsymbol{\phi}^{\text{istrap}}$.
```

V. Bootstrap Planning

This section introduces an MPC-based path planner to solve Problem 2. First, we reformulate the obstacle avoidance

[2] Note Assumption 1 **does not** imply full state measurement.

[3] The parameters δ_t and N^σ are dictated by measurement noise levels, which can be characterized off-line in a controlled experimental setting.

[4] The effectiveness of Algorithm 1 depends highly on the time delay length L, the number of training measurements N^{train}, the number of bootstraps N^{strap}, and the MPC horizon length, N^h. We recommend that N^{train} be at least $10N^h$ and that $L = \frac{N^{\text{train}}}{4}$. N^{strap} and N^{step} should be as large as allowed by the computing platform and benchmarking them offline.

constraint (6d) in terms of the mean and variance of the bootstrap predictions. Next, we use this constraint in the MPC optimization, and provide probabilistic guarantees of constraint satisfaction. Algorithm 1 produces N^{strap} copies of N^h length predictions of the k^{th} obstacle's location. We denote the j^{th} copy of the bootstrap prediction as $\{\hat{\boldsymbol{y}}_k^j\}_1^{N_h} = \{\hat{\boldsymbol{y}}_{1,k}^j, \hat{\boldsymbol{y}}_{2,k}^j, \cdots, \hat{\boldsymbol{y}}_{N^h,k}^j\}$. The collision avoidance set constraint (6d) can be reformulated based on the obstacle shape and center as $\|C\boldsymbol{x}_i - \hat{\boldsymbol{y}}_{i,k}^j\|_2 \geq \hat{r}_k + r_p \triangleq \overline{r}_k$, for each $i \in \mathbb{Z}^{1:N^h}$, $k \in \mathbb{Z}^{1:N^{\text{obs}}}$, and where r_p is the agent's safety radius (4). This collision constraint can be reexpressed as the following concave (in the state $\boldsymbol{x}_i$) constraint,

$$(C\boldsymbol{x}_i - \hat{\boldsymbol{y}}_{i,k}^j)^T(C\boldsymbol{x}_i - \hat{\boldsymbol{y}}_{i,k}^j) \geq \overline{r}_k\|(C\boldsymbol{x}_i - \hat{\boldsymbol{y}}_{i,k}^j)\|_2. \quad (8)$$

We approximate (8) as an affine constraint through the use of Sequential Convex Programming (SCP) [15], [16]

$$(C\boldsymbol{x}_i - \hat{\boldsymbol{y}}_{i,k}^j)^T(C\overline{\boldsymbol{x}}_i - \hat{\boldsymbol{y}}_{i,k}^j) \geq \overline{r}_k\|(C\overline{\boldsymbol{x}}_i - \hat{\boldsymbol{y}}_{i,k}^j)\|_2, \quad (9)$$

where $\overline{\boldsymbol{x}}_i$ is approximated with the solution from previous SCP iterations. Note that (9) over-approximates constraint (8) [15].

Lemma 2. *If we have N^{strap} forecasts of thekth obstacle's position from time $i \in \mathbb{Z}^{1:N^h}$ and the previous SCP trajectory $\{\overline{\boldsymbol{x}}\}_1^{N_h}$, then we can define the j^{th} bootstrap lumped collision avoidance coefficients $\alpha_{i,k}^j$, $\beta_{i,k}^j$ and the standard deviation of the collision avoidance constraint $\Delta_{i,k}$ as:*

$$\alpha_{i,k}^j \triangleq -C^T(C\overline{\boldsymbol{x}}_i - \hat{\boldsymbol{y}}_{i,k}^j) \quad (10)$$

$$\beta_{i,k}^j \triangleq \overline{r}_k\|(C\overline{\boldsymbol{x}}_i - \hat{\boldsymbol{y}}_{i,k}^j)\|_2 - (C\overline{\boldsymbol{x}}_i)^T(C\overline{\boldsymbol{x}}_i - \hat{\boldsymbol{y}}_{i,k}^j) \quad (11)$$

$$\Delta_{i,k} \triangleq \sqrt{\boldsymbol{p}_i^T\Sigma_{\alpha_{i,k}}\boldsymbol{p}_i + 2\boldsymbol{p}_k^T\Sigma_{\alpha\beta_{i,k}} + \Sigma_{\beta i,k}}, \quad (12)$$

where, $\Sigma_{\alpha_{i,k}} \triangleq cov\left(\alpha_{i,k}^j, \alpha_{i,k}^j\right)$, $\Sigma_{\beta_{i,k}} \triangleq cov\left(\beta_{i,k}^j, \beta_{i,k}^j\right)$, and $\Sigma_{\alpha\beta_{i,k}} \triangleq cov\left(\alpha_{i,k}^j, \alpha_{i,k}^j\right)$ are sample covariance matrices computed using the bootstrapped coefficients $\{\alpha_{i,k}\}_1^{N^{strap}}$ and $\{\beta_{i,k}\}_1^{N^{strap}}$ and $\boldsymbol{p}_i \triangleq C\boldsymbol{x}_i \in \mathbb{R}^3$. Let the dimension of the null space of $\Sigma_{\alpha_{i,k}}$ be$n_{i,k} \geq 0$.[5] The standard deviation $\Delta_{i,k}$ has the following upper bound,

$$\Delta_{i,k} \leq \mathbf{1}^T|\tilde{\Sigma}_{\alpha_{i,k}}^{\frac{1}{2}}(\boldsymbol{p}_i - \boldsymbol{h}_{i,k})| + \sqrt{3k_{i,k}} \triangleq \zeta_{i,k}, \quad (13)$$

where $\tilde{\Sigma}_{\alpha_{i,k}} = \Sigma_{\alpha_{i,k}} + I_{i,k}^{null}$, $I_{i,k}^{null} = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & I_{n_{i,k}\times n_{i,k}} \end{bmatrix} \in \mathbb{R}^{3\times 3}$, and

$$\begin{bmatrix} \boldsymbol{h}_{i,k} \\ k_{i,k} \end{bmatrix} \triangleq \begin{bmatrix} -\left(\Sigma_{\alpha_{i,k}} + I_{i,k}^{null}\right)^{-1}\Sigma_{\alpha\beta_{i,k}} \\ \Sigma_{\beta_{i,k}} - \Sigma_{\alpha\beta_{i,k}}^T\left(\Sigma_{\alpha_{i,k}} + I_{i,k}^{null}\right)^{-1}\Sigma_{\alpha\beta_{i,k}} \end{bmatrix}. \quad (14)$$

Proof. See [21]. □

It is costly to incorporate each bootstrap as a separate obstacle constraint, as the number of constraints grow linearly with N^{strap}. Instead, we estimate the ensemble mean and covariance of the distance from the obstacle. A DRCC accounts for *all bootstrap distributions* that can have this mean and covariance. This approach results in a significantly fewer obstacle constraints, whose cardinality remains fixed regardless of the number of bootstrap predictions N^{strap}.

Theorem 3. *(SSA-MPC) Consider Problem 2 under Assumptions 1-3 with system dynamics* (4) *and bootstrap SSA forecasts of all obstacles' center positions. Given a risk tolerance ϵ, the solution to the following optimal control problem is a feasible solution of Problem 2 as $w \to \infty$. The SCP optimization problem at iteration w is:*

$$\{\boldsymbol{u}^*\}_1^{N_h} = \min_{\substack{\boldsymbol{u}_i \in \mathbb{R}^{n_{\boldsymbol{u}}} \\ \boldsymbol{s}_{i,k} \in \mathbb{R}^3}} \sum_{i=1}^{N_h} L_i(\boldsymbol{y}_i^{ref} - G\boldsymbol{x}_i, \boldsymbol{u}_i) \quad (15a)$$

$$s.t. \quad \boldsymbol{x}_{i+1} = A\boldsymbol{x}_i + B\boldsymbol{u}_i \quad (15b)$$

$$\boldsymbol{x}_i \in \mathcal{D}^{\boldsymbol{x}}, \quad \boldsymbol{u}_i \in \mathcal{D}^{\boldsymbol{u}}, \quad \boldsymbol{x}_1 = \boldsymbol{x}_{init} \quad (15c)$$

$$\Lambda_{i,k}\begin{bmatrix}\boldsymbol{x}_i & \boldsymbol{s}_{i,k}\end{bmatrix}^T \leq \Gamma_{i,k} \quad (15d)$$

$$\|\boldsymbol{x}_i - \overline{\boldsymbol{x}}_i\| \leq \chi\tau^w \;\; \forall(i,k) \in \mathbb{Z}^{1:N^h} \times \mathbb{Z}^{1:N^{obs}} \quad (15e)$$

where $\{\bar{\boldsymbol{x}}\}_1^{N^h}$ is the solution to the $(w-1)^{th}$ iteration of the SCP optimization, $\Lambda_{i,k} \in \mathbb{R}^{7\times 11}$ and $\Gamma_{i,k} \in \mathbb{R}^7$ encode the risk-based collision avoidance relationships,

$$\Lambda_{i,k} = \begin{bmatrix} \mathbb{E}[\alpha_{i,k}]^T C & \mathbf{1}^T\nu_{\epsilon_n} \\ \tilde{\Sigma}_{\alpha_{i,k}}^{1/2} C & -I_{3\times 3} \\ -\tilde{\Sigma}_{\alpha_{i,k}}^{1/2} C & -I_{3\times 3} \end{bmatrix}, \quad \Gamma_{i,k} = \begin{bmatrix} -\mathbb{E}[\beta_{i,k}] - \nu_{\epsilon_n}\sqrt{3k_{i,k}} \\ \tilde{\Sigma}_{\alpha_{i,k}}^{1/2}\boldsymbol{h}_{i,k} \\ -\tilde{\Sigma}_{\alpha_{i,k}}^{1/2}\boldsymbol{h}_{i,k} \end{bmatrix},$$

where $\epsilon_n \triangleq \frac{\epsilon}{N^{obs}}$ and $\nu_{\epsilon_n} \triangleq \sqrt{\frac{1-\epsilon_n}{\epsilon_n}}$. Lastly, $\chi \geq 0$ is the initial trust region and $\tau \in (0,1)$ the worst-case rate of convergence.

Proof. Denote the j^{th} random bootstrapped obstacle forecastsas $\boldsymbol{z}_{i,k}^j \triangleq (\alpha_{i,k}^j)^T\boldsymbol{x}_i + \beta_{i,k}^j$, where $\alpha_{i,k}^j$ and $\beta_{i,k}^j$ are defined in (10) and (11). The obstacle avoidance constraint (8) has an affine over approximation (9), which is equivalently given by $\boldsymbol{z}_{i,k}^j < 0$. Hence, the chance constraint (6d) is,

$$\mathbb{P}(\boldsymbol{x}_i \in \mathcal{O}_{\text{pred}}) = \mathbb{P}\Big(\bigcup_{k=1}^{N^{\text{obs}}}\{\boldsymbol{z}_{i,k} \geq 0\}\Big) \leq \sum_{k=1}^{N^{\text{obs}}}\mathbb{P}(\boldsymbol{z}_{i,k} \geq 0).$$

Enforcing $\mathbb{P}(\boldsymbol{z}_{i,k} \geq 0) \leq \epsilon_n$, $\forall k \in \mathbb{Z}^{1:N^{\text{obs}}}$ also satisfies (6d). We can satisfy this constraint as a DRCC:

$$\sup_{\boldsymbol{\kappa}\sim\left(\mathbb{E}[\boldsymbol{z}_{i,k}], \Sigma_{\boldsymbol{z}_{i,k}}\right)} \mathbb{P}\{\boldsymbol{\kappa} \geq 0\} \leq \epsilon_n, \; \forall i,k \in \mathbb{Z}^{1:N^h} \times \mathbb{Z}^{1:N^{\text{obs}}},$$

where $\mathbb{E}[\boldsymbol{z}_{i,k}]$ and $\Sigma_{\boldsymbol{z}_{i,k}}$ are the sample mean and covariance of the bootstrapped $\{\boldsymbol{z}_{i,k}\}_1^{N^{\text{strap}}}$. We reformulate the above statement as a deterministic constraint as shown in [22],

$$\underbrace{\mathbb{E}[\boldsymbol{z}_{i,k}]}_{\mathbb{E}[\alpha_{i,k}]^T C\boldsymbol{x}_i + \mathbb{E}[\beta_{i,k}]} + \nu_{\epsilon_n}\underbrace{\sqrt{\Sigma_{\boldsymbol{z}_{i,k}}}}_{\Delta_{i,k}} \leq 0, \; \forall i \in \mathbb{Z}^{1:N^h}, k \in \mathbb{Z}^{1:N^{\text{obs}}}. \quad (16)$$

Constraint (16) is not affine in the optimization variable, as is desirable for real-time application. By Lemma 2, $\Delta_{i,k} \leq \zeta_{i,k}$, and we deduce the following tighter inequality constraint as a numerically appealing alternative to (16),

$$\mathbb{E}[\alpha_{i,k}]^T C\boldsymbol{x}_i + \mathbb{E}[\beta_{i,k}] + \nu_{\epsilon_n}\left(\mathbf{1}^T|\Sigma_{\alpha_{i,k}}^{1/2}\boldsymbol{p}_i - \boldsymbol{h}_{i,k}| + \sqrt{3k_{i,k}}\right) \leq 0. \quad (17)$$

To account for the absolute value term, we introduce auxiliary optimization variables $\boldsymbol{s}_{i,k}$ that satisfy the following:

$$\Sigma_{\alpha_{i,k}}^{1/2}\boldsymbol{p}_i - \boldsymbol{h}_{i,k} \leq \boldsymbol{s}_{i,k}, \quad -\Sigma_{\alpha_{i,k}}^{1/2}\boldsymbol{p}_i + \boldsymbol{h}_{i,k} \leq \boldsymbol{s}_{i,k}.$$

Therefore, satisfying (15d) is equivalent to satisfying (17).

[5] In all numerical simulations, $\Sigma_{\alpha_{i,k}}$ is strictly positive definite. However, when one or more measurable states are noiseless, $\Sigma_{\alpha_{i,k}}$ can be ill-conditioned. Instead of adding $I_{i,k}^{\text{null}}$, which can be numerically expansive to determine, we recommend applying Algorithm 1 only to states with measurement noise, and adapt Theorem 3 with deterministic forecasts for the noiseless states and the DRCC formulation for the noisy ones.

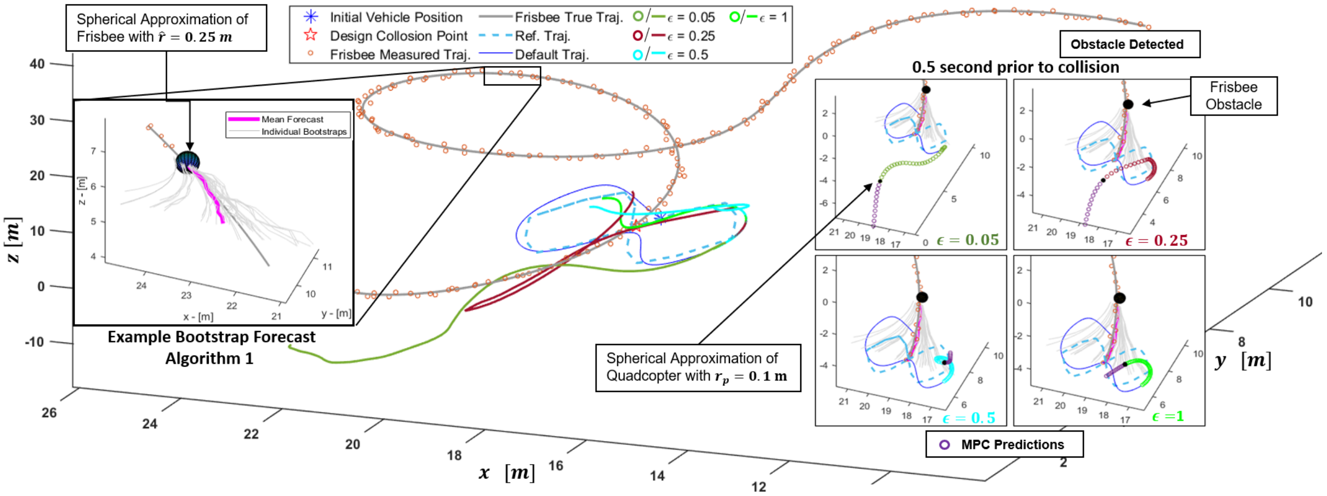

Fig. 2: Four Monte-Carlo simulations with agent dynamics (18) and a Frisbee obstacle (see Fig.1) are compared. The same obstacle behaviors are simulated while the agent tracks the same figure '8' reference trajectory with four risk levels $\epsilon = \{0.05, 0.25, 0.5, 1\}$. The simulation is designed to be difficult: the vehicle must deviate from its reference trajectory as the obstacle trajectory is designed to intersect the agent's reference trajectory with noise obstacle trajectory measurements. All measurement noises are sampled uniformly between $[-0.125,\ 0.125]$ meters. The bootstrap obstacle forecast uses the parameters: $L = 24$, $N^{\text{train}} = 100$, $N^{\text{step}} = 5$, $\delta_t = 20$, $N^{\sigma} = 8$, $N^{\text{strap}} = 40$. SSA-MPC uses the constants $N^h = 10$, $\chi = 50$ and $\tau = 0.25$ with fixed 4-step SCP iterations. The tuple $(\{\lambda^j\}_1^{t_j}, \{\boldsymbol{\mu}^j\}_1^{t_j}, \phi^j), \forall j \in \mathbb{Z}^{1:40}$ in Algorithm 1 is computed with observables measured at 20 Hz. The four sub-diagrams show the planned trajectory at 4 risk levels; the planner is more conservative as $\epsilon \to 0$, and aligns with the results shown in Table I and II.

Cases	ϵ	0.05	0.1	0.25	0.5	0.75	1
Const. Speed	%Feas.	97.5	98.2	98.9	99.6	99.9	100
	%Succ.	100	100	100	100	100	59.0
	$\bar{d}_{min}$	2.26	1.85	1.41	1.12	0.94	0.64
	$\sigma(d_{min})$	0.42	0.33	0.25	0.22	0.24	0.35
Ball w/drag	%Feas.	99.5	99.6	99.9	100	100	100
	%Succ.	100	100	100	100	100	79.3
	$\bar{d}_{min}$	2.60	2.14	1.63	1.27	1.04	0.64
	$\sigma(d_{min})$	1.08	0.93	0.70	0.50	0.37	0.27
Frisbee w/drag	%Feas.	90.3	97.4	98.3	98.6	97.5	97.8
	%Succ.	100	100	100	100	99.7	58.0
	$\bar{d}_{min}$	4.97	3.97	2.85	2.01	1.44	0.78
	$\sigma(d_{min})$	1.97	1.53	1.15	0.91	0.76	0.77

TABLE I: Summary of results from MC simulations of system (18)

Cases	ϵ	0.05	0.1	0.25	0.5	0.75	1
Const. Speed	%Feas.	83.9	84.2	85.6	86.5	88.6	99.8
	%Succ.	100	100	100	100	100	61.2
	$\bar{d}_{min}$	10.42	8.67	6.59	4.88	3.57	0.46
	$\sigma(d_{min})$	2.47	2.28	1.86	1.43	1.16	0.14
Ball w/drag	%Feas.	92.8	91.8	90.9	91.1	90.7	82.8
	%Succ.	100	100	100	100	100	0.1
	$\bar{d}_{min}$	6.95	5.82	4.25	3.04	2.13	1.82
	$\sigma(d_{min})$	2.52	2.55	2.27	1.98	1.59	N/A
Frisbee w/drag	%Feas.	92.2	94.5	92.9	92.3	86.9	100
	%Succ.	100	100	100	100	100	40.5
	$\bar{d}_{min}$	12.5	11.3	9.10	6.59	4.24	0.34
	$\sigma(d_{min})$	2.47	2.84	3.00	2.81	2.27	0.05

TABLE II: Summary of results from MC simulations of system (19)

Convergence of the SCP is proven in [23] which is based on implementing a trust region via second-order cone constraints (15e). The authors also show the solution to the SCP as $w \to \infty$ is a feasible solution to Problem 2.[6] □

VI. Numerical Examples

We consider a quadcopter that follows a reference trajectory $\boldsymbol{y}^{\text{ref}}$ while avoiding unknown moving obstacles and respecting state and control constraints. Its position and Euler angles (roll, pitch, yaw) in frame E are denoted $x, y, z,\ \varphi, \theta, \psi$.

Example 1: Fully-actuated Multirotor with attitude controller: Assume there exists a low level attitude controller that tracks given attitude commands within 20 Hz. Therefore, we use the following linear dynamic model to extract a high-level motion planner that outputs attitude and thrust inputs:

$$\ddot{x} = -g\theta,\ \ddot{y} = g\varphi,\ \ddot{z} = u_1 - g,\ \dot{\psi} = u_4, \tag{18}$$

where the planner control inputs are given by $u_1, \theta, \varphi, u_4$ which are thrust, roll angle, pitch angle, and yaw rate, and where $g = 9.81 m/s^2$ is the gravitational acceleration.

Example 2: Multirotor operating in small angle regime: A mixer maps thrust and moment inputs into electronic speed controller PWM commands at 8kHz. Since the multirotor is constrained to operate within the state constraints $\theta \in [-0.45, 0.45]$ radians and $\varphi \in [-0.45, 0.45]$ radians, we use the following standard multirotor linear dynamics,

$$\ddot{x} = -g\theta,\ \ddot{y} = g\varphi,\ \ddot{z} = u_1 - g, \tag{19}$$

$$\ddot{\varphi} = \frac{u_2}{I_{xx}},\ \ddot{\theta} = \frac{u_3}{I_{yy}},\ \dot{\psi} = \frac{u_4}{I_{zz}}, \tag{20}$$

where the planner control inputs u_1, u_2, u_3, u_4 correspond to the thrust force in the body frame and three moments. The vehicle's moments of inertia are $I_{xx} = 0.0075 kgm^2, I_{yy} = 0.0075 kgm^2, I_{zz} = 0.013 kgm^2$. For both examples, the desire reference trajectory consists of positions, $\{x^{\text{ref}}\}, \{y^{\text{ref}}\}, \{z^{\text{ref}}\}$ and yaw angles $\{\psi^{\text{ref}}\}$.

We conducted Monte-Carlo (MC) simulations of our planner as it avoids three differently behaved obstacles which are introduced once in each run for both examples. See code.[7] Case 1 uses a constant speed spherical obstacle without drag. Case 2 is a thrown spherical obstacle with drag. In Case 3, a Frisbee is thrown at various initial positions, velocities, and rotation speed. The sphere dynamics are captured by a 6-state ODE with drag penalties proportional to its velocities. The Frisbee is modeled following [17], using a 12-state model

[6] To be numerically feasible, w is usually upper bounded by a finite integer, resulting in a sub-optimal but still feasible solution.

[7] https://github.com/skylarXwei/Riskaware_MPC_SSA_Sim.git

identical to Fig. 1 with aerodynamic drag. The Frisbee is modeled as a sphere with the same radius as the Frisbee disk. We also benchmark our method against an artificial potential field alternative. A supplementary video (`https://youtu.be/6s8pfRZ171Q`) provides more details. We conducted 1000 MC simulations per ϵ level to compare the numerical feasibility, percent success in obstacle avoidance (if the MPC planner is feasible), and the planner's conservativeness, as measured by the minimum distance between the obstacle and agent centers. For the three cases, the obstacle speed ranges are $[0.41,\, 8.43]$, $[3.41,\, 6.37]$, and $[5.76,\, 6.68]$ m/s, respectively. The MPC planning and measurement rates are fixed at 20 Hz. With a 10 step horizon and 40 bootstraps, the average per planner update rate is 0.030 ± 0.0014 sec, using Gurobi [24] on an Intel i7-9700K CPU @3.6GHz processor, using a dynamic simulation written in MATLAB. The results in Tables I and II show the applicability of our SSA-MPC algorithm, despite vast differences in obstacle behavior. Further, as the risk tolerance ϵ shrinks, the percentage success in obstacle avoidance (when the solution is feasible) increases, with a trade-off in the feasibility of optimization (15). Parameters d_{min} and $\sigma(d_{min})$ are the average minimum distance between the agent and the obstacle and the standard deviation of this minimum distance across 1000 MC simulations, respectively. Based on Table I and II, the risk tolerance ϵ can also viewed as a robustness parameter which is inversely proportional to the distance between the agent and obstacles. However, the cost of more robustly avoiding the obstacles is reflected in the numerical feasibility, a parameter describing the chances of the SCP formulation (15) being feasible for the entire simulation. The feasible set of the polytopic collision avoidance constraints (15d) shrinks as ϵ (and hence ϵ_n) decreases.

There are limitations of our approach. It degrades as more noise is injected into the system, which calls for more robust denoising methods than hard thresholding. The horizon length also matters. We observed that insufficient denoising may lead to a poor bootstrap forecast that compounds with increasing horizon length. A similar phenomenon is also observed in [25]. While risk-aware MPC handles a poor forecast internally, it could lead to numerical infeasibility.

VII. Conclusion and Future Work

Our data-driven, risk-aware obstacle avoidance planner showcased near perfect results in avoiding moving obstacles with limited, noisy measurements and no prior knowledge about the obstacle behaviors. We offer a new paradigm that can extract obstacle dynamics online in reasonable time and a compatible risk-aware MPC formulation that also works in real-time. However, we discovered limitations in the current formulation that suggest future work. As a major limitation, our use of a linear MPC formulation significantly restricts the multirotor from performing agile maneuvers. Appealing to nonlinear model predictive methods could reduce the numerical infeasibility due to roll and pitch state constraints. In terms of computational efficiency, the number of auxiliary optimization variable $\boldsymbol{s}_{i,k}$ and the number of collision avoiding inequality constraints all scale linearly with the number of obstacles. There is significant room for improvement of this efficiency by assigning obstacle priorities. Our approach also needs to be benchmarked in high fidelity environments like GAZEBO and/or validated by hardware experiments in the future. The simulation results show that adjusting the risk level, which is also the probability of constraint violation, ϵ can implicitly adjust the safety distance between the agent and obstacles. This observation can be further strengthened by empirical correlations and/or theoretical results.

References

[1] D. Fan, K. Otsu, Y. Kubo, A. Dixit, J. Burdick, and A.-A. Agha-Mohammadi, "Step: Stochastic traversability evaluation and planning for risk-aware off-road navigation," in *Robot.: Sci. Syst.*, 2021.

[2] J. Latombe, *Robot motion planning*. Kluwer, 1996.

[3] J. Canny, *The complexity of robot motion planning*. MIT press, 1988.

[4] C. Tomlin, G. Pappas, and S. Sastry, "Conflict resolution for air traffic management: a study in multiagent hybrid systems," *IEEE Transactions on Automatic Control*, vol. 43, no. 4, pp. 509–521, 1998.

[5] J. van den Berg, D. Ferguson, and J. Kuffner, "Anytime path planning and replanning in dynamic environments," in *IEEE ICRA*, 2006.

[6] C. Lam, C. C. K. Chiang, and L. Fu, "Human-centered robot navigationtowards a harmoniously humanrobot coexisting environment," *IEEE Trans. Robotics*, vol. 27, no. 1, pp. 99–112, 2011.

[7] B. Lindqvist, S. Mansouri, A. Agha, and G. Nikolakopoulos, "Nonlinear mpc for collision avoidance and control of uavs with dynamic obstacles," *IEEE Robot. & Autom. Lett*, 2020.

[8] I. Hwang and C. Seah, "Intent-based probabilistic conflict detection for the next generation air transportation system," *Proceedings of the IEEE*, vol. 96, no. 12, pp. 2040–2059, 2008.

[9] S. Brunton, B. Brunton, J. Proctor, E. Kaiser, and J. Kutz, "Chaos as an intermittently forced linear system," *Nat. commun.*, 2017.

[10] F. Takens, "Detecting strange attractors in turbulence," in *Dynamical systems and turbulence*, pp. 366–381, Springer, 1981.

[11] N. Goljandina and A. Zigljavskij, *Singular spectrum analysis for time series*. Springer, 2020.

[12] N. Golyandina and A. Korobeynikov, "Basic singular spectrum analysis and forecasting with r," *Comp. Stat. & Data Anal.*

[13] J. Juang and R. Pappa, "An eigensystem realization algorithm for modal parameter identification and model reduction," *J. Guidance, Control, and Dynamics*, vol. 8, no. 5, pp. 620–627, 1985.

[14] U. Fasel, J. Kutz, B. Brunton, and S. Brunton, "Ensemble-sindy: Robust sparse model discovery in the low-data, high-noise limit, with active learning and control," *Proc. R. Soc. A*, vol. 478, no. 2260, 2022.

[15] D. Morgan, S. Chung, and F. Hadaegh, "Model predictive control of swarms of spacecraft using sequential convex programming," *Journal of Guidance, Control, and Dynamics*, vol. 37, pp. 1–16, 04 2014.

[16] F. Augugliaro, A. Schoellig, and R. D'Andrea, "Generation of collision-free trajectories for a quadrocopter fleet: A sequential convex programming approach," *Int. Conf Intell. Robots Syst.*, 2012.

[17] S. Hummel, *Frisbee flight simulation and throw biomechanics*. University of California, Davis, 2003.

[18] H. Adbi and L. Williams, "Principal component analysis," *Wiley interdisc. reviews: comp. statistics*, vol. 2, no. 4, pp. 433–459, 2010.

[19] N. Golyandina, V. Nekrutkin, and A. Zhigljavsky, "Analysis of time series structure - ssa and related techniques," in *Monographs on statistics and applied probability*, 2001.

[20] A. Agarwal, A. Alomar, and D. Shah, "On multivariate singular spectrum analysis," *arXiv preprint arXiv:2006.13448*, 06 2020.

[21] S. X. Wei, A. Dixit, S. Tomar, and J. W. Burdick, "Moving obstacle avoidance: a data-driven risk-aware approach," *arXiv preprint arXiv:2203.14913*, 2022.

[22] L. Ghaoui, M. Oks, and F. Oustry, "Worst-case value-at-risk and robust portfolio optimization: A conic programming approach," *Operations Research*, vol. 51, pp. 543–556, 08 2003.

[23] R. Foust, S. Chung, and F. Hadaegh, "Optimal guidance and control with nonlinear dynamics using sequential convex programming," *J. of Guid., Control, and Dyn.*, vol. 43, no. 4, pp. 633–644, 2020.

[24] Gurobi Optimization, LLC, "Gurobi Optimizer Ref. Manual," 2022.

[25] B. Lindqvist, P. Sopasakis, and G. Nikolakopoulos, "A scalable distributed collision avoidance scheme for multi-agent uav systems," in *2021 IROS*, pp. 9212–9218, IEEE, 2021.