Optimal Linear Multiple Estimation for Landmark-Based Planning via Control Synthesis

Chenfei Wang and Roberto Tron

Abstract—A common way to implement navigation in mobile robots is through the use of landmarks. In this case, the main goal of the controller is to make progress toward a goal location (stability), while avoiding the boundary of the environment (safety). In our previous work, we proposed a method to synthesize global controllers for environments with a polyhedral decomposition; our solution uses a Quadratically Constrained Quadratic Program with Chance Constraints to take into account the uncertainty in landmark measurements. Building upon this work, we introduce the concept of virtual landmarks, which are defined as linear combinations of actual landmark measurements that minimize the uncertainty in the resulting control actions. Interestingly, our results show that the first minimum-variance landmark is independent of the feedback control matrix, thus decoupling the design of the landmark from the one of the controller; attempting to derive additional, statistically independent landmarks, however, requires solving non-convex problems that involve also the controller. Numerical experiments demonstrate that, by minimizing the variance of the inputs, the resulting controller will be less conservative, and the robot is able to complete navigation tasks more effectively (faster and with less jitter in the trajectory).

I. INTRODUCTION

Motion planning is a fundamental problem for mobile robotics. A basic approach [9], [18], [23] consists of abstracting the environment into a graph (e.g., via a cell decomposition or random sampling), and then dividing the planning problem into a high-level planning on the vertices of the graph (i.e., sequences of regions) followed by a low-level feedback controller for transitions (between regions). In this setting, the high-level planning component is relatively simple, and can be completed, for instance, with the celebrated A* algorithm. The low-level controller, however, needs to balance multiple requirements and challenges: making progress toward the next step of the high-level plan toward a goal (stability), avoiding the environment boundary (safety), reduce the effect of noise in the measurements (robustness to noise). One way to address this set of requirements is by focusing on linear feedback laws, and perform a worst-case analysis to synthesize controllers that are stable and safe for given probabilistic bounds on the noise in the system, as done in our previous work [26]. This approach results in a minimal amount of computations at runtime (which simply involves applying the pre-computed linear controller), but, by its nature, will result in conservative control actions.

In this paper, however, we make the following observations:

(O1) We can shape the uncertainty of feedback control actions (to a certain degree) through different linear combinations of the inputs.

(O2) The probability of violating a safety constraint, i.e., moving toward a wall, depends on the uncertainty of the control along the normal to the wall (moving parallel to the wall does not increase the risk).

(O3) In practice, statistically consistent estimators (e.g., Kalman filters [7]) include a covariance matrix for the uncertainty in the estimate at each time step; this information could be used in real time to reduce the conservatism of a worst-case analysis.

A. Background and Previous work

Stability and safety are crucial for control system. Control Lyapunov Functions (CLFs, [1], [25]) and Control Barrier Functions (CBFs, [2], [3]) have become a popular methods to create controllers that ensure, respectively, user-defined convergence guarantees and forward invariance of safety sets; this popularity is due to the flexibility they provide (they only require mild conditions on the differentiability of the barrier functions and can be applied to systems with high relative degree [13], [19], [29]), and the fact that can be implemented relatively efficiently, as they generally result in quadratic programs (QPs). While typically employed for real-time control, CLFs and CBFs have also been applied to offline path planning. In particular, the present paper follows the paradigm introduced by the work of [4], [5], which uses linear CLF and CBF functions to design linear output feedback controllers that can guide the robot in a convex region, using high-level discrete path planning to determine a sequence of regions toward the goal. In our previous work [27], we have extended the same paradigm to the case where the measurements are affected by Gaussian noise with known bounds on the covariance. In this case, the CLF and CBF constraints become stochastic quantities that can be transformed into chance CLF and CBF constraints (ensuring that the original CLF and CBF constraints hold with a user-defined probability) using Chebyshev’s bound; this formulation leads to solving Quadratic Constraint Quadratic Programings (QCQPs) instead of QPs. Although the resulting controllers are a significant improvement with respect to other work that does not explicitly consider noise (e.g., [6], [10], [15]), they can still be overly conservative: for instance, [27] does not provide a way to incorporate better bounds if these become available at runtime.

The effect of noise (and other disturbances) can be also considered on the controls (i.e., the controller outputs a
nominal $u$, but the system receives $u$ plus a disturbance $d$; this can be taken into account again by using a worst-case analysis leading to robust QP formulations [8], [11], [14], [16], [21], [24] or by considering steady-state-errors [17]. Such line of work, however, considers disturbances affecting the output of the controller (Fig. 1a), while we consider noise affecting the input of the controller (Fig. 1b); the two settings are somewhat complementary, and we will focus on the latter.

B. Paper Contributions

In this work, we build upon observations (O1)–(O3) to enhance our prior work [26]. We first discuss a novel way to use the linearity of the controllers to mix the different input landmark measurements (Section II). Then, we show that we can reduce the conservatism of the chance CBF constraints by creating virtual landmarks that minimize the variance in the direction of the constraints (Sec. III-C). We show that it is possible to generate at most $N_y$ uncorrelated virtual marks from $N_y$ original landmarks (Sec. III-B). Interestingly, our analysis shows that the first virtual landmark can be selected independently of the linear feedback controller and the safety constraints, while the subsequent ones require solving a joint non-convex problem over both the combination weights and the controller. The first virtual landmark can be computed by from the covariance matrices using simple matrix operations. Hence, we propose therefore the following strategy: offline, we use the known covariance bounds to merge all landmarks into a single virtual landmark (Fig. 1c), and synthesize controllers based on the tightened chance constraints; online, if updated covariance matrices are available, we can recompute the virtual landmark and update the controller with simple matrix operations. Both steps reduce the conservatism of the control actions.

We show via numerical simulations that the reduced conservatism translates in a final trajectory that converges faster and with less jitter.

II. PROBLEM FORMULATION AND PRELIMINARIES

A. Dynamical System

We model the robot as a linear time-invariant system

$$\dot{x} = Ax + Bu,$$

where $A \in \mathbb{R}^{n \times n}$, and $B \in \mathbb{R}^{n \times m}$ are assumed to form a stabilizable pair, $x \in \mathbb{R}^n$ is the state, and $u \in \mathbb{R}^m$ is the control input (the output measurement model is also linear and is discussed in §II-C). Our formulation heavily relies on the linearity of model (1). However, it can be applied to nonlinear robots by either linearizing the model (note that each environment cell can use a different linearization point) or by using a lower-level nonlinear controller (e.g., see [5] for a practical demonstration of this approach).

B. The Environment and Decomposition-Based Navigation

It is assumed that the free space in the environment can be decomposed into convex cells (polytopes). As in [4], we tackle the path planning problem by first executing a high level path planning strategy, which identifies a sequence of cells from the one that contains the starting location to the one that contains the goal location. This is typically implemented by first representing the cells and the possible motions between them as the vertices and edges of a graph, respectively, and then using a discrete path planning algorithm such as $\mathcal{A}^*$. Then, we design controllers for each cell $\mathcal{X}$ in the decomposition that use measurements with respect to fixed landmarks, as explained in the two sections below. Each cell $\mathcal{X}$ is defined by a set of linear inequalities

$$\mathcal{X} = \{x \in \mathbb{R}^n : A_x x + b_x \leq 0\},$$

where each row of the system $A_x x + b_x$ defines the hyperplane of one of the faces of the polytope.

C. Landmark Measurement Model

We assume that, for each cell $\mathcal{X}$, the locations of $N_y \geq 2$ landmarks $\{Y_i\}_{i=1}^L$ are known at planning time (if $N_y = 1$ it is still possible to solve the path planning problem, but there is no opportunity to apply the main results of this paper discussed in Sec. III-A). We assume that the robot is able to measure the relative displacements (see Fig. 2)

$$y_i(x) = Y_i - x + \theta_i,$$

with noise $\theta_i \sim \mathcal{N}(0, \Sigma_i)$. We assume that each measurement $y_i$ is statistically independent from other measurements $y_j$, $j \neq i$. In practice, the measurements can be acquired using

![Fig. 2: Landmark based robot navigation: the robot can measure the relative location of landmarks. And it utilizes this information to complete the control and navigation task.](image)
typical hardware installed on mobile robots, such as LIDARs or RGBD cameras, possibly integrating multiple measurements over time using SLAM [12] or filtering techniques [22]. To simplify the notation for the derivations in the next section, we write (3) in vector form as $y = Y + \theta$, where $y = \operatorname{stack}\{y_i\}$, $Y = \operatorname{stack}\{Y_i\}$, and $\theta = \operatorname{stack}\{\theta_i\}$.

Since the location of the landmarks is known and the measurement model (3) is known, it is possible to simulate the measurement of landmarks with arbitrary locations, as detailed in the following.

**Definition 1:** We define a virtual landmark as an arbitrary location $Y'$ obtained as a linear combination of the existing landmarks, i.e., $Y' = \sum_{i}^{N_y} W_i Y_i$, where $\sum_{i}^{N_y} W_i = 1$.

**Proposition 1:** With the notation above, the virtual measurement $y_W = Y_W - x$ corresponding to the virtual landmark $Y_W$ is computed as a linear combination of the measurements, $y_W = \sum_{i}^{N_y} W_i y_i + \theta_W$, where $\theta_W = \sum_{i}^{N_y} W_i \theta_i$ is a noise vector with distribution $\mathcal{N}(0, \Sigma_W)$, where $\Sigma_W = \sum_{i}^{N_y} W_i \Sigma_i W_i^T$.

**Proof:** The first part of the claim follows from the definitions of $y_W$ and $Y_W$ and the distributivity property: $y_W = \sum_{i}^{N_y} W_i y_i = \sum_{i}^{N_y} W_i Y_i + \sum_{i}^{N_y} W_i x + \sum_{i}^{N_y} W_i \theta_i = Y_W - x + \theta_W$. The second part of the claim derives from the fact that the mean of $\theta_W$ is $\mathbb{E}[\theta_W] = \sum_{i}^{N_y} W_i \mathbb{E}[\theta_i] = 0$, and its covariance is $\mathbb{E}[\theta_W \theta_W^T] = \sum_{i}^{N_y} W_i \mathbb{E}[\theta_i \theta_i^T] W_i^T = \sum_{i}^{N_y} W_i \Sigma_i W_i^T$, since $\theta_i, \theta_j$ are statistically independent for $i \neq j$.

**D. Landmark-Based Linear Controllers**

Once the sequence of cells to traverse is established, we need a local controller that can guide the robot during the transition from a cell to the following one. In our approach we define a linear feedback controller that combines the multiple measurements $\{y_i\}$ according to

$$u = \sum_{i}^{N_y} K_i y_i + k = K Y + k,$$  

(4)

where $K_i$ is the feedback matrix for measurement $y_i$, $k$ is a common control bias term, and $N_y$ is the number of measurements; additionally, the controller is written in vectorized form using $\mathbf{K} = \begin{bmatrix} K_1 & \ldots & K_{N_y} \end{bmatrix}$. Note that (4) represents an output feedback controller, because we do not measure the state $x$ directly, but $N_y$ affine versions of it, each one with, in general, different noise characteristics.

We now introduce a preliminary result which allow to compute (4) using different expressions that, in the absence of noise, are all equivalent.

**Proposition 2 (Modified from [5]):** Let $Y_W \in \mathbb{R}^d$ be a virtual landmark. Then, the control $u$ in (4) can be computed from the virtual measurement $y_W = Y_W - x$ as

$$u = K_W y_W + k_W,$$  

(5)

where $K_W = \sum_{i}^{N_y} K_i$ and $k_W = \sum_{i}^{N_y} K_i Y_i - Y_W$.

See [5] for a proof. The proposition, in principle, allows us to compute the control $u$ based on any point in the environment (and in particular with respect to arbitrary virtual landmarks).

**E. Control Barrier Function**

Control Barrier Functions (CBFs) are a widely used tool in safety critical system.

**Definition 2 (ECBF, [20]):** For an affine nonlinear system $\dot{x} = f(x) + g(x)u$ and a set $\mathcal{C} = \{x \in \mathbb{R}^n : h(x) \geq 0\}$, where the system has relative degree $r$ with respect to the function $h(x)$, $h(x)$ is an **Exponential Control Barrier Function** (ECBF [20]), if there exists $\alpha_h \in \mathbb{R}^{1 \times r}$ such that

$$L_f h(x) + L_g L_f^{-1} h(x) u + \alpha_h \xi(x) \geq 0,$$  

(6)

where $\xi(x) = [h(x), L_f h(x), ..., L_f^{r-1} h(x)]^T$ and $\alpha_h = [\alpha_{h0}, \alpha_{h1}, ..., \alpha_{hr-1}]$. The row vector $\alpha_h$ is selected satisfying the requirements given by [20].

In the context of a robot navigation task, a wall or boundary of a convex region is represented with the barrier function $h(x) = A_h x + b_h$, where $A_h \in \mathbb{R}^{1 \times n}$ represents the normal to the wall, $b_h \in \mathbb{R}$, and the set $\mathcal{C} = \{x \in \mathbb{R}^n : h(x) > 0\}$ represents the free space outside the wall. Considering the LTI system (1), inequality (6) can be written in matrix form as

$$A x + B u + d \geq 0,$$  

(7)

where

$$A = \begin{bmatrix} A_h & A_h A_r^{-1} & \vdots & A_h \end{bmatrix}, \quad B = \begin{bmatrix} A_h \end{bmatrix}, \quad d = \alpha_h b_h.$$

**Remark 1:** For simplicity, we present our theory using a single CBF constraint. However, it is straightforward to extend the results to multiple the general case with multiple walls by using additional constraints of the form (7).

**F. Control Barrier Function Chance Constraints**

In practice, the control output $u$ for constraint (7) is calculated from the measurements as in (4) or (5), when virtual landmarks are used, which contains noise and is hence a stochastic quantity. As a consequence, the inequality constraint (7) becomes a stochastic quantity as well: we therefore rewrite it as a chance constraint

$$P(A x + B u + d \geq 0) \geq 1 - \eta(x),$$  

(8)

where $P$ represents a Gaussian distribution, and $\eta(x)$ is a function of the state $x$ satisfying the following conditions:

- $0 < \eta(x) < 1$ for $x \in \mathcal{X}$,

- $\eta(x)$ is continuous and concave.

At this point, we could use the results of [27] to transform (8) into constraints that are quadratic in $K$ and linear in $x$ (so that they can be used to find the controllers); however, we propose here a derivation based on the Gaussian cumulative distribution function that is both simpler and leads to tighter constraints.

**Proposition 3:** A sufficient condition for (8) to hold is

$$K_2 \Gamma K_2^T - K_1 x - K_3 \leq 0$$  

(9)
where \( \eta_0 \) is a lower bound on \( \eta \) in the current region \( \mathcal{X} \), \( \Gamma = \frac{1}{\eta_0} \Sigma \), and
\[
\bar{K}_1 = A - B \sum_i K_i, \quad (10)
\]
\[
\bar{K}_2 = BK, \quad (11)
\]
\[
\bar{K}_3 = B(KY + k) + d. \quad (12)
\]
Before proving the proposition, we need the following:

**Lemma 1:** Let \( \Phi(\cdot) \) be the cumulative distribution of a scalar Gaussian random variable with zero mean and unit variance. The following inequality holds for \( \eta > 0 \):
\[
\frac{1}{\eta} > \Phi^{-1}(1-\eta). \quad (13)
\]

**Proof:** Consider the function \( f(\eta) = \Phi(\frac{1}{\eta}) + \eta - 1, \eta > 0 \).
We have \( \lim_{\eta \to 0^+} f(\eta) = 0 \) and \( \frac{d}{d\eta} f(\eta) = 1 - \frac{2e^{-\eta^2}}{\sqrt{\pi} \eta^3} \geq 1 - \frac{2}{e^2 \sqrt{\pi}} > 0 \). Thus, we get \( f(\eta) = \Phi(\frac{1}{\eta}) + \eta - 1 > 0 \), from which the claim follows.

**Proof:** Substituting (4) into (8) we obtain
\[
P(\bar{K}_1 x + \bar{K}_2 \theta + \bar{K}_3 \bar{y} \geq 0) \geq 1 - \eta(x). \quad (14)
\]

Letting \( \Sigma = \text{blkdiag}\{(\Sigma_i)\} \), we have that the random variables \( \frac{1}{\sqrt{K_2 \Sigma K_2^T}} \bar{K}_2 \theta \) have zero mean and unit variance. From the definition of cumulative distribution, we then have that (14) is equivalent to
\[
\frac{\bar{K}_1 x + \bar{K}_3}{\bar{K}_2 \Sigma K_2^T} \geq \Phi^{-1}(1-\eta(x)). \quad (15)
\]

Furthermore, from Lemma 1 and the definition of \( \eta_0 \), we have
\[
\frac{1}{\eta_0} \geq \frac{1}{\eta(x)} > \Phi^{-1}(1-\eta(x)). \quad (16)
\]
Combining (15) with (16) and rearranging terms, we obtain the result of the proposition.

**Remark 2:** The term \( \sigma^2 \bar{K} = K_2 \Sigma K_2^T \) indicates the variance of the controller towards the direction of the barrier boundary. From (15), we see that reducing \( \sigma^2 \bar{K} \) makes the chance constraint looser, hence giving the opportunity to improve \( K \). In Sec. III, we show how to minimize this variance \( \sigma^2 \bar{K} \) (and hence improve \( K \)) by weighting the measurements.

We can apply (9) as a constraint while optimizing, e.g., the Frobenius norm of the feedback matrix, leading to the optimization problem
\[
\min ||K||_2^2
\]
subject to \( \bar{K}_2 \Gamma \bar{K}_2^T - \bar{K}_1 x - \bar{K}_3 \leq 0 \quad \forall x \in \mathcal{X} \quad (17) \]
The optimization can be rewritten as a QCQP (see [28] for a detailed derivation).
\[
\min ||K||_2^2
\]
subject to \( \bar{K}_2 \Gamma \bar{K}_2^T - \bar{K}_3 + b \lambda \leq 0 \)
\[
A \lambda + \bar{K}_1^T = 0
\]
\[
\lambda \geq 0
\]

An analogous process (omitted here for the sake of brevity), can be applied to obtain Control Lyapunov Function chance constraints that can be incorporated in the optimization problem (18): it is sufficient to replace \( A_h, b_h \) with the definition of the hyperplane identifying the exit face of the cell, inverting the sense of the inequality in (7), and then follow the same steps.

### III. Virtual Landmarks for Minimum Chance Control Synthesis

In this section we propose to improve the feasibility of the QCQP (18) by replacing the measurements and landmarks \( \{y_i, Y_i\} \) with virtual landmarks. We first consider the case of a single virtual landmark (for which the optimal weights \( \{W_i\} \) can be obtained independently of the controller \( K \) and the wall parameters \( A_h, b_h \), and then the case of multiple, statistically independent landmarks (which require solving for the weights \( \{W_i\} \) and the controller \( K \) simultaneously).

#### A. Single Virtual Landmark

If we use a single measurement \( y_W \) with respect to a single virtual landmark \( Y_W \) (see Def. 1), the optimization problem (18) is still valid, except that \( \bar{K}_1, \bar{K}_2 \) and \( \bar{K}_3 \) need to be changed; as a consequence, from the characterization of the covariance of \( y' \) in Prop. 1, we have that \( \sigma^2_k = \bar{K}_2 \Sigma W \bar{K}_2^T \). Following Remark 2 we propose to find the set of weights \( \{W_i\} \) that maximize the feasibility of the QCQP (18); specifically, we aim to solve the problem
\[
\min_i \bar{K}_2 \Sigma W \bar{K}_2^T
\]
subject to \( \sum_i W_i = I \) \quad (19)

**Proposition 4:** The solution of the optimization problem 19 can be obtained with simple matrix operations:
\[
W_i = \left( \sum_i \Sigma_i^{-1} \right)^{-1} \Sigma_i^{-1} \quad (20)
\]
The proof of this proposition is given in [28], and is related to standard proofs for Weighted Least Squares estimation.

**Remark 3:** The matrix \( \bar{K}_2 \) does not appear in (20). Thus, the weight \( W_i \) is independent with the controller \( K \). This shows that the control design can be factorized into optimal estimation and control, which is reminiscent of LQG design.

**Remark 4:** It is possible to prove that \( \sigma^2_K \) cannot become worse after fusing the measurements into a single landmark, although such proof has been omitted due to space constraints.

#### B. Linear Multiple Estimation

In the previous section we have shown how to create a single virtual landmark from the initial \( N_y \) landmarks to improve the feasibility of the design problem for the controller \( K \); the solution, as it turns out, does not depend on the actual controller chosen. Intuitively, however, the use of a single landmark may loose some information due to the reduction in dimensionality. It is then natural to ask whether it is possible to find additional virtual landmarks that may provide complementary information (more precisely, that are statistically uncorrelated). In this section we show that this is
indeed possible, although the result depends on the controller (which needs to be synthesized jointly).

Inspired by the Principal Component Analysis (PCA) which decomposes the original data to different directions, we propose a similar decomposition-based weighting method.

Suppose we want to generate a set of additional virtual measurements \( l > 1 \).

\[
y^{W,l} = \sum_i W_i^l y_i,
\]

where \( y^{W,k} \) indicates the \( k \)-th weighted measurement. To ensure that each weighted measurement carries novel information, we require that it is uncorrelated with previous virtual landmarks by enforcing the constraint

\[
\mathbb{E}[y_i y_j^T] = 0.
\]

For \( l = 1 \), \( \{W_i^1\} \) is given by (20); \( y_i^1 = \sum_i W_i^1 y_i \), our first virtual landmark, can be interpreted as the first principal component. For \( l \geq 2 \), \( \{W_i^l\} \) needs to follow the uncorrelated constraint (22), which can be rewritten as the following form,

\[
\sum_i \left( W_i^l \right)^T \Sigma_i W_i^l = 0
\]

To show this, suppose that the constraint (23) hold. For two virtual landmarks \( j \) and \( l \), the noise terms are \( \delta_j = \sum_i W_i^j \theta_i \) and \( \delta_l = \sum_i W_i^l \theta_i \). Requiring that \( y_j \) and \( y_l \) are uncorrelated can be formally stated as:

\[
\text{Cov}(\delta_j, \delta_l) = \mathbb{E}[\delta_j \delta_l^T]
\]

\[
= \mathbb{E} \left[ \left( \sum_i W_i^j \theta_i \right) \left( \sum_i W_i^l \theta_i \right)^T \right]
\]

\[
= \sum_i \sum_i W_i^j \mathbb{E}(\theta_i \theta_i^T) W_i^l^T
\]

\[
= \sum_i W_i^j \Sigma_i W_i^l^T = 0,
\]

With the constraints above, we can find each new landmark \( y_i^l \) in sequence after solving for \( y_i^1, \ldots, y_i^{l-1} \) by solving the following optimization problem (which is similar to (19) with the additions of the uncorrelation constraints (23)),

\[
\min_{\{W_i^l\}} K_2 \sum_i \left( W_i^l \Sigma_i W_i^l^T \right) K_2^T
\]

s.t. \( \sum_i W_i^l = I \)

\[
\sum_i (W_i^l)^T \Sigma_i W_i^l = 0,
\]

\[
\sum_i (W_i^{l-1})^T \Sigma_i W_i^l = 0,
\]

The solution is given in [28].

The weights for virtual landmarks other than the first, i.e., \( l > 1 \), depend on \( K_2 \), which contains the controller \( K \) (as shown in (25) and already mentioned above). Hence, to rigorously make use of additional virtual landmarks, we would need to solve (18) jointly with the weights \( W_i^l \), under the constraints from (25). In general, this results in a non-convex formulation, and whose solution is beyond the scope of this paper. As a result, we will limit the simulations to the case of a single landmark, \( l = 1 \), showing that the this already significantly improves the performance of the controller.

C. Control Synthesis

As discussed above, for each chance constrain (i.e., for each wall), we can pick virtual landmarks that minimize the variance in each chance CBF constraint. To actually synthesize the controller \( K \), we therefore propose the following steps:

1) Generate virtual landmarks \( y_{W,l} \).
2) Solve the controller design problem (18) based on the virtual landmarks.
3) If the noise covariance information is updated, use simple matrix operations to recompute the weights (Prop. 4 or eq. 20) and then adapt the controller using (2).

Since the weight of the first virtual landmark is unrelated to the controller, it can always be calculated before the feedback controller design. Hence, the controller synthesis should consider the noise and covariance of the first virtual landmark (Prop. 1) instead of noise of the physical landmarks.

IV. NUMERICAL VALIDATION AND DISCUSSION

In this section, we first demonstrate the application of Linear Multiple Estimation for multiple landmarks; then, we show the effectiveness the feedback control synthesis with a single virtual landmark against multiple physical landmarks. For all the plots, the blue (respectively, red) markers indicate physical (respectively, virtual) landmarks with randomly-generated positions; the corresponding ellipses represent the covariance of the measurements; solid straight lines indicates the boundaries of the environment.

Fig. 3 shows the result of using LME to obtain the top three virtual landmarks for a boundary. The coordinate of the three virtual landmarks are shown. The left case and the right case have different feedback matrices \( K \) which leads to different \( K_2 \).

A. Virtual Landmarks for a Given Controller

Fig. 3 shows the result of using LME to obtain the top three virtual landmarks for two different controllers \( K \) (an the corresponding \( K_2 \)). As expected, the first landmark is the same between the two cases, and has the smallest covariance. The remaining virtual landmarks' covariance is minimized in a specific direction which is determined by the direction of the boundary and the controller together.

B. Effectiveness of Virtual-Landmark-Based Control

Fig. 4 compares control synthesis with physical landmarks and a single virtual landmark (\( l = 1 \)) in a corridor-like environment. The simulation is based on a robot with single-integrator dynamics \( \dot{x} = u \) and uses an Euler–Maruyama
scheme with time step 0.1 s. The robot inputs to the robot are represented by the relative displacement to the landmarks; for the sake of simplicity, we consider constant covariance matrices for the measurement noise (i.e., which does not depend on time and position); note that, in this simulation, the covariance is large enough to be comparable to the width of the corridor. At each time step, the input $u$ is calculated using the controller $K$ computed from $(18)$.

The controller that uses physical landmarks successfully avoids collisions with the walls, but the trajectory exhibits some chatter, and the control actions are more conservative, leading to a total time of 27.7 s to exit the corridor. The controller that uses the first virtual landmark (whose covariance is around one-third of each one of the physical landmarks) results in a trajectory with less jitter, with more aggressive control actions, reducing the total travel time to 8.6 s.

V. CONCLUSIONS

In this paper, we propose the Linear Multiple Estimation (LME) method for landmark-based path planning and control, which generates uncorrelated virtual landmarks based on physical landmarks. The analytical results show that the first virtual landmark computed with LME is a weighted average of the physical landmarks with weights that are independent of the controller (which can then be designed at the same time) and the environment. Successive (statistically independent) landmarks instead depend on the controller used and the environment; nonetheless, the LME formulation provides a way to compute optimal weights for a given controller.

The numerical experiments shows that, even when a single landmark is used, the proposed strategy significantly reduces the effect of noise and improves the convergence speed.

Avenues for future work include a rigorous experimental evaluation on a real robot, using Kalman filters for the measurements and their time-variant covariance estimates, using LME to adjust the weights in real time.

REFERENCES

[1] A. D. Ames, K. Galloway, K. Sreenath, and J. W. Grizzle. Rapidly exponentially stabilizing control lyapunov functions and hybrid zero dynamics. IEEE Transactions on Automatic Control, 59(4):876–891, 2014.

[2] A. D. Ames, J. W. Grizzle, and P. Tabuada. Control barrier function based quadratic programs with application to adaptive cruise control. In CDC, pages 6271–6278. IEEE, 2014.

[3] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada. Control barrier function based quadratic programs for safety critical systems. IEEE Trans. Autom. Control., 62(8):3861–3876, 2017.

[4] M. Bahreinian, E. Aasi, and R. Tron. Robust path planning and control for polygonal environments via linear programming. In IEEE American Control Conference, 2021.

[5] M. Bahreinian, M. Mitjans, and R. Tron. Robust sample-based output-feedback path planning. In IEEE International Conference on Intelligent Robots and Systems, 2021.

[6] C. Bela, V. Ileri, and G. J. Pappas. Discrete abstractions for robot motion planning and control in polygonal environments. IEEE Transactions on Robotics and Automation, 21(5):864–874, 2005.

[7] S. Y. Chen. Kalman filter for robot vision: A survey. IEEE Transactions on Industrial Electronics, 59(11):4409–4420, 2012.

[8] J. J. Choi, D. Lee, K. Sreenath, C. J. Tomlin, and S. L. Herbert. Robust control barrier–value functions for safety-critical control. In IEEE International Conference on Decision and Control, pages 6814–6821, 2021.

[9] H. M. Choset, K. M. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L. Kavraki, and S. Thrun. Principles of robot motion: theory, algorithms, and implementation. MIT press, 2005.

[10] D. C. Conner, A. A. Rizzi, and H. Choset. Composition of local potential functions for global robot control and navigation. In IEEE International Conference on Intelligent Robots and Systems, volume 4, pages 3546–3551, 2003.

[11] C. Dawson, Z. Qin, S. Gao, and C. Fan. Safe nonlinear control using robust neural Lyapunov-barrier functions. In Conference on Robot Learning, pages 1724–1735. PMLR, 2022.

[12] E. Eade and T. Drummond. Scalable monocular SLAM. In IEEE Conference on Computer Vision and Pattern Recognition, volume 1, pages 469–476, 2006.

[13] K. Galloway, K. Sreenath, A. D. Ames, and J. W. Grizzle. Torque saturation in bipedal robotic walking through control lyapunov function-based quadratic programs. IEEE Access, 3:323–332, 2015.

[14] K. Garg and D. Panagou. Robust control barrier and control lyapunov functions with fixed-time convergence guarantees. In IEEE American Control Conference, pages 2292–2297, 2021.

[15] L. Habets, P. J. Collins, and J. H. van Schuppen. Reachability and control synthesis for piecewise-affine hybrid systems on simplices. IEEE Transactions on Automatic Control, 51(6):938–948, 2006.

[16] M. Jankovic. Robust control barrier functions for constrained stabilization of nonlinear systems. Automatica, 96:359–367, 2018.

[17] S. Kolathaya and A. D. Ames. Input-to-state safety with control barrier functions. IEEE Control Systems Letters, 3(1):108–113, 2018.

[18] S. M. LaValle. Planning algorithms. Cambridge university press, 2006.

[19] Q. Nguyen and K. Sreenath. Exponential control barrier functions for enforcing high relative-degree safety-critical constraints. In IEEE American Control Conference, pages 322–328, IEEE, 2016.

[20] Q. Nguyen and K. Sreenath. Exponential control barrier functions for enforcing high relative-degree safety-critical constraints. In ACC, pages 322–328, IEEE, 2016.

[21] Q. Nguyen and K. Sreenath. Robust safety-critical control for dynamic robotics. IEEE Transactions on Automatic Control, 2021.

[22] R. J. Prazenica, A. Watkins, A. J. Kurtdila, Q. F. Ke, and T. Kanade. Vision-based kalman filtering for aircraft state estimation and structure from motion. In AIAA Guidance, Navigation, and Control Conference, volume 5, page 3, 2005.

[23] A. Schürmann. Computational geometry of positive definite quadratic forms. University Lecture Series, 49, 2009.

[24] P. Seiler, M. Jankovic, and E. Hellstrom. Control barrier functions with unmodeled input dynamics using integral quadratic constraints. IEEE Control Systems Letters, 6:1664–1669, 2021.

[25] E. D. Sontag. A ‘universal’construction of artstein’s theorem on nonlinear stabilization. Systems & control letters, 13(2):117–123, 1989.

[26] C. Wang, M. Bahreinian, and R. Tron. Chance constraint robust control with control barrier functions. In 2021 American Control Conference (ACC), pages 2315–2322, 2021.

[27] C. Wang, M. Bahreinian, and R. Tron. Chance constraint robust control with control barrier functions. In IEEE American Control Conference, 2021.

[28] C. Wang and R. Tron. Technical report on optimal linear multiple estimation for landmark-based planning via control synthesis. ArXiv, 2018.

[29] W. Xiao and C. Belta. High order control barrier functions. IEEE Transactions on Automatic Control, 2021.