Sampling-Based Robust State Estimation for Deterministic Safety Controllers

Marfred Barrera
Department of Mechanical and Aerospace Engineering, University of California San Diego
Email: mrbarrera@ucsd.edu

***Abstract*—Established safety frameworks such as Hamilton-Jacobi Reachability and Control Barrier Functions provide powerful formal guarantees for performance and safety of dynamical systems. However, they traditionally assume perfect knowledge of the state and deterministic state transitions. In the case that there is process or measurement uncertainty within the system, these formal guarantees are compromised.**

To address this limitation, this project leverages a Nested Monte Carlo method from stochastic Model Predictive Control. This approach selects the best-performing state estimate from candidates generated by a particle filter by optimizing a prescribed finite-horizon cost function while satisfying state and input constraints. The state estimate that minimizes the cost is then used in Certainty Equivalence feedback control. When combined with a Quadratic Program-based Control Barrier Function safety filter, this heuristic approach to stochastic optimal control provides a robust state estimate, improving the safety of deterministic controllers in stochastic settings while preserving performance objectives.

Computational examples are included to demonstrate the performance of the State Selection Algorithm applied to different sets of dynamics and controllers, and opportunities for future work are discussed.

***Index Terms*—Stochastic Optimal Control, Stochastic MPC, State Estimation, Particle Filtering, Reachability, Control Barrier Functions**

I. Introduction

With the widespread growth and adoption of autonomous systems in everyday life, the need for advances in performance and safety becomes essential. Although liveness has been at the forefront of control, guarantees for safety grow increasingly central to the success of autonomous systems, which must tackle increasingly complex environments, especially in safety-critical scenarios such as autonomous driving and collision avoidance. Safe control algorithms such as Hamilton-Jacobi Reachability [1] and Control Barrier Functions [2] provide powerful formal guarantees regarding the safety of dynamical systems, but they traditionally assume deterministic state transitions and measurements, which means that there are no random disturbances and there is no difference between open-loop and closed-loop trajectories. Evidently, the real world is not deterministic, so making this assumption is a rarely applicable luxury. In the event that there is process or measurement uncertainty within the system, the equivalence between open-loop and closed-loop control disappears, and the formal safety guarantees of Hamilton-Jacobi Reachability and Control Barrier Functions are undermined if not properly addressed.

On the other hand, the addition of uncertainty or partial observability into the formulation dips into the field of stochastic optimal control, which can quickly become intractable [3]. Combined with the already restrictive computational complexities of some techniques for safe control, there exists proper motivation to look for a compromise between complete optimality and the convenience of a sub-optimal, heuristic approach to stochastic optimal control.

Toward this end, previous work in [4] presented a method for stochastic Model Predictive Control which attempts to find this middle ground between complete optimality and tractability in stochastic systems through Monte Carlo sampling. Monte Carlo sampling-based methods are widely used in stochastic optimization because they enable numerical approximations without requiring closed-form solutions, and they are particularly useful for approximating models that are difficult to evaluate. A survey of Monte Carlo sampling for estimation and optimization is given in [5].

Using techniques from nested Monte Carlo, [3] demonstrates an approach to stochastic Model Predictive Control that selects the best performing state among candidates generated from a particle filter for use in Certainty Equivalence feedback control. The best state is found by minimizing a prescribed cost function over the set of state estimates while also satisfying state and input constraints by sampling control sequences and their corresponding open-loop trajectories. Through this State Selection Algorithm, we can influence the behavior of the particle density towards some control objective through the selection of the state rather than the selection of the control signal itself. By selecting the state estimate whose sampled trajectories minimize a performance function and also satisfies state and input constraints, this method of state estimation enforces safety as a hard constraint while also providing room to optimize a given performance metric.

In this project, we aim to combine the State Selection Algorithm with methods for safe control such as Hamilton-Jacobi reachability and control barrier functions to improve the performance and safety of traditionally deterministic safety algorithms when applied to stochastic settings. The combination of these two techniques would be an approach to stochastic optimal control that explores the compromise between complete optimality and tractable, heuristic techniques while incorporating an additional layer of robustness, and it is a contribution to the study of state estimation within control barrier functions and reachability.

II. Problem Statement

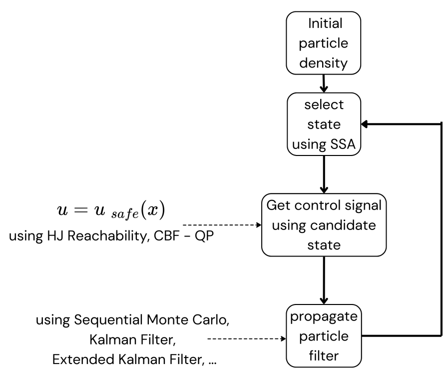

Fig. 1: Problem Formulation with SSA and a safe controller

For this project, we will first demonstrate the State Selection Algorithm using a simple numerical example. Then, we combine the SSA with a Control Barrier Function problem. Specifically, we analyze an obstacle avoidance problem for a double integrator where the control input is acceleration in the x and y directions. In a deterministic setting, a Quadratic Programming - Control Barrier Function safety filter provides a formal guarantee for safety, but on top of this CBF problem we add random disturbance and measurement noise, which eliminates the guarantees for safety. Now that the problem is in a stochastic setting, we produce a probability distribution of the state with a particle filter. Using this particle density, we compare the performance and safety of a nominal conditional-mean-based Certainty Equivalence state estimate and the State Selection Algorithm.

A. State Selection Algorithm Formulation

Consider the following dynamical system:

I. Discrete-time state dynamics

$$x_{k+1} = f(x_k, u_k, w_k) \qquad (1)$$
$$y_k = h(x_k, v_k)$$

where state $x_k \in \mathbb{R}^n$, control input $u_k \in \mathbb{R}^m$, disturbance $w_k \in \mathbb{R}^d$, and noise $v_k \in \mathbb{R}^e$

II. Stochastic disturbance process, $\{w_k\}$, assumed to be independently and identically distributed (i.i.d) possessing known density $\mathcal{W}$. The state x_0 is independent of w_k for all k.

III. State and input constraint sets, $\mathbb{X}$ and $\mathbb{U}$, respectively

IV. Nominal full-state-feedback control law $u_k = \kappa(x_k)$

V. An inital state x_0-density p_0, provided as a collection of particles $\Xi = \{\xi_0^i \in \mathbb{R}^n, i = 1, ..., L\}$

VI. An N-stage finite-horizon trajectory cost function $J = \sum_{k=1}^{N} \ell_k(x_k, u_k)$

The goal of the State Selection Algorithm is to select a candidate state value, $x_0^\star$ from the initial state density which is probabilistically feasible (Monte Carlo language for "safe") and favorably influences the average trajectory cost, J, over the set of particles. As displayed in Figure 1, this candidate state is fed into a nominal controller, and the control value is applied to the system.

In this project, the discrete time dynamics are of a 4 dimensional double integrator with an output that detects position, each with additive Gaussian noise,

$$x_k = \begin{bmatrix} p_{x,k} \\ p_{y,k} \\ v_{x,k} \\ v_{y,k} \end{bmatrix}, \quad y_k = \begin{bmatrix} p_{x,k} \\ p_{y,k} \end{bmatrix}, \quad u_k = \begin{bmatrix} \dot{v}_{x,k} \\ \dot{v}_{y,k} \end{bmatrix} \qquad (2)$$

and the running-cost function is an LQR based quadratic cost

$$\ell(x, u) = x'Qx + u'Ru \qquad (3)$$

where $Q = I$ and $R = 2I$.

B. Control Barrier Function Formulation

The nominal controller for this project is a Quadratic Programming-Control Barrier Function derived safety filter, which unifies performance objectives with safety requirements. We give an abridged definition of a control barrier function below with a more rigorous formulation available in [6].

Consider a safe set $\mathcal{C}$ defined as the superlevel set of a continuously differentiable function $h : D \subset \mathbb{R}^n \to \mathbb{R}$.

$$C = \{x \in D \subset \mathbb{R}^n : h(x) \geq 0\} \tag{4}$$

Then h is a control barrier function if there exists an extended class $\mathcal{K}_\infty$ function α such that for a control system $\dot{x} = f(x) + g(x)u$,

$$\sup_{u \in U}[L_f h(x) + L_g h(x)u] \geq -\alpha(h(x)) \tag{5}$$

for all $x \in D$ [6].

Other common alternative representations of the safety constraint described by (5) include

$$\begin{aligned} \dot{h}(x) + \alpha(h(x)) &\geq 0 \\ \ddot{h}(x) + c_1\dot{h}(x) + c_2 h(x) &\geq 0 \end{aligned} \tag{6}$$

depending on the relative degree between the control input and h. Any control value that satisfies (5) or (6) guarantees invariance within the safe set $\mathcal{C}$.

To modify a given feedback controller $u_{nom} = \kappa(x)$ with safety guarantees, we can formulate a Quadratic Program based controller:

$$\begin{aligned} u_{safe}(x) = \arg\min_{u \in \mathbb{R}^m} & \; ||u - u_{nom}(x)||^2 \\ s.t. \quad & \ddot{h}(x) + c_1\dot{h}(x) + c_2 h(x) \geq 0 \end{aligned} \tag{7}$$

This QP step incorporates both hard safety constraints and performance objectives into the controller.

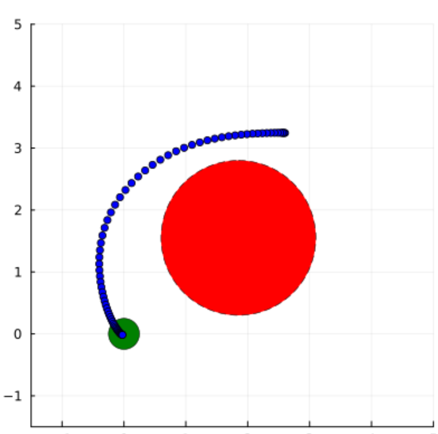

Fig. 2: Deterministic QP-CBF obstacle avoidance

For this project, we define a control barrier function $h(x)$ that encodes a safe set $\mathcal{C}$ and unsafe set $\mathcal{C}^c$ as

$$h(x) = distance(x, r_{obs}) - d_{min} \tag{8}$$

where r_{obs} is the position of the center of a circular obstacle, and d_{min} is the radius of the obstacle. When combined with the State Selection Algorithm, this problem's unsafe set $\mathcal{C}^c$ is equivalent to the state constraint set $\mathbb{X}$.

Additionally, the nominal control law is defined as a PD controller that stabilizes the double integrator towards the origin.

$$u(x) = -K_d[v_x, v_y]^T - K_p[p_x, p_y]^T \tag{9}$$

III. Technical Approach

The structure of a combined State Selection Algorithm and QP-CBF, which is briefly expressed in Figure 1, is outlined as follows:

1) Define initial state density Ξ_0
2) Perform the Certainty Equivalence step using the State Selection Algorithm, narrowing the probability distribution to a single particle $x^\star$
3) use $x^\star$ in feedback controller, $u_{safe} = \kappa(x^\star)$
4) Propagate the distribution using the state transition model
5) Update the distribution using a particle filter

By keeping to this general structure, various types of feedback control, particle filtering, and cost function minimization can be used, making this SSA technique modular and adaptable. Next, a more thorough explanation of the State Selection Algorithm, QP-Control Barrier Function, and Particle Filtering steps is provided.

Algorithm 1 Outer loop of SSA-QP-CBF

Input: State density Ξ, particle filter (PF), state selection algorithm (SSA), state transition model $f(x, u, w)$, QP-CBF-based controller $u_{safe} = \kappa(x)$
Initialization: initial state density Ξ_0
Output: propagated state density

```
for k = 1, 2, ... do
    Perform SSA; x* ← arg min J(x)
    Compute control value; u ← κ(x*)
    Propagate Ξ_k; x_{k+1} ← f(x_k, u_k, w_k)
    Update likelihoods; p_0 ← P(y|x)
    if Particle Depletion then
        Resample
    end if
end for
```

A. State Selection Algorithm

Say we start with an initial state density Ξ_0 with L number of particles. The algorithm proceeds as follows:

1. Select a Monte Carlo sample number M and statistical feasibility tolerance $\alpha \in [0, \epsilon)$, which are parameters of the algorithm
2. For each $i \in \{1, 2, ..., L\}$, choose $x'_0 = \xi^i \in \Xi$.
   i) sample from $\mathcal{W}$ and Ξ independent realizations of noise sequences w'_j and w''_j, as well as initial states $x''_{0,j}$
   ii) compute the N-long κ-closed-loop state sequence from x'_0.
   $$x'_{k+1,j} = f(x'_{k,j}, \kappa(x'_{k,j}), w'_{k,j}) \tag{10}$$
   including the corresponding closed-loop control sequence $\{\kappa(x'_{k,j})\}_{k=0}^{N-1}$
   iii) For each $j \in \{1, ..., M\}$, compute the open-loop-controlled state sequence from sampled states $x''_{0,j}$,
   $$x''_{k+1,j} = f(x''_{k,j}, \kappa(x'_{k,j}), w''_{k,j}) \tag{11}$$
   iv) Compute for each k the sample-average closed-loop control violation rate,
   $$\hat{\beta}_k(x'_0) = \frac{1}{M} \sum_{j=1}^{M} \mathbb{1}(\kappa(x''_{k,j}) \in \mathbb{U}). \tag{12}$$
   v) Compute for each k the sample-average open-loop-controlled state violation rate
   $$\hat{\lambda}_k(x'_0) = \frac{1}{M} \sum_{j=1}^{M} \mathbb{1}(x''_{k,j} \in \mathbb{X}). \tag{13}$$
   vi) If $\hat{\beta}_k(x'_0) \geq 1 - \alpha$ and $\hat{\lambda}_k(x'_0) \geq 1 - \alpha$ for all k, then declare this state x'_0 to be feasible
   vii) If x'_0 is feasible, calculate the sample-average performance
   $$J_c^M(x'_0) = \frac{1}{M} \sum_{j=1}^{M} \sum_{k=0}^{N} l_k(x''_{k,j}, \kappa(x'_{k,j})). \tag{14}$$
3. Pick $x_0^\star$ to be the feasible x'_0 minimizing $J_c^M(\cdot)$
   $$x_0^\star = \arg \min_{x'_0 \in \mathbb{X}_0^\epsilon} J_c(x'_0) \tag{15}$$

In summary, for each particle $x'_0 \in \Xi$, we generate M trajectories using the particle's closed-loop control sequence $\{u_{k,j} = \kappa(x'_{k,j})\}$ and take the sample averages of the open-loop cost of the closed-loop sequence. Then, the selected state is the state of lowest cost that also belongs to the feasible set $\mathbb{X}^\epsilon$.

Algorithm 2 State Selection Algorithm

Input: state density Ξ, state transition model $f(x, u, w)$, running-cost function $l(x, u)$, nominal controller $u = \kappa(x)$
Output: candidate state $x^\star$

```
for i = 1, 2, ..., L do
    calculate x' sequence, x'_{k+1} ← f(x'_k, κ(x'_k), w'_k)
    record control sequence, u ← κ(x'_k)
    sample M number of states from Ξ, x''_k
    for j = 1, 2, ..., M do
        calculate x'' sequence using u
            x''_{k+1} ← f(x''_k, κ(x'_k), w''_k)
        α' ← checkViolation(x''_k)
    end for
    calculate average cost of x'' sequences,
        J(x'_k) ← (1/M) ΣΣ l(x''_k, u_k)
    if α' < α then
        add x'_k to feasible set X^ε, X^ε ← X^ε ∪ {x'}
    end if
end for
    return x* ← arg min J(x')
```

The State Selection Algorithm as described above was recreated from scratch as part of a Julia program with a particle density size $L = 400$ and a Monte Carlo sample number $M = 135$. Initially, a basic numerical example was used to demonstrate the capabilities of the SSA by itself. Once this numerical example was established, the SSA was combined with the QP-CBF problem.

B. Control Barrier Function

Once the candidate state $x^\star$ is found, it is used in a nominal feedback controller $u = \kappa(x)$. For this project, the feedback controller is the QP-CBF-based safety filter defined in equation (7).

Because the control input enters the dynamics further down the derivative chain compared to position-based control barrier function, the higher order CBF constraint defined in (6) is used. Since these dynamics are affine in control, the CBF constraint can be restructured into the form $Au \geq b$, and the control law turns into a quadratic programming optimization problem. This optimization problem is solved using Julia's JuMP optimization package, resulting in a safety-filtered control value that is used to propagate the probability distribution.

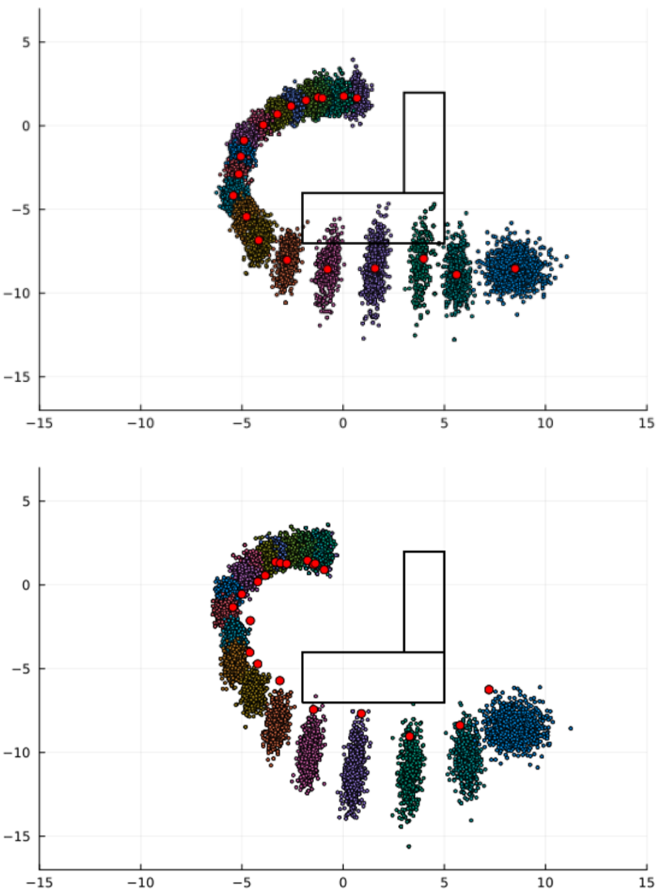

Fig. 3: Initial numerical example of the State Selection Algorithm in Julia

C. *Particle Filter*

Once the distribution goes through the state transition, a measurement is taken of the simulated true state (including additive Gaussian noise) for use in a bootstrap particle filtering update.

This particle filter step is also implemented from scratch in Julia, which goes as follows:

1. Propagate the particle density

$$x_{k+1,i} = f(x_{k,i}, u_k, w_k) \tag{16}$$

2. Propagate the simulated true state and take measurement, including additive Gaussian noise

$$y_{k+1} = g(x_{k+1}, v_{k+1}) \tag{17}$$

3. Calculate the weights associated with each particles using the likelihood of the observation,

$$w_{k+1,i} = w_{k,i}\ p(y_{k+1}|x_{k+1,i}) \tag{18}$$

then normalize

$$w_{k+1}^{(i)} = \frac{w_{k+1}^{(i)}}{\sum_{i=1}^{L} w_{k+1}^{(i)}} \tag{19}$$

4. If the density size is below a certain threshold, resample particles and reset weights to 1.

Now that there is an updated particle density, the process starts over, and the Certainty Equivalence step with SSA can be performed again.

IV. Results

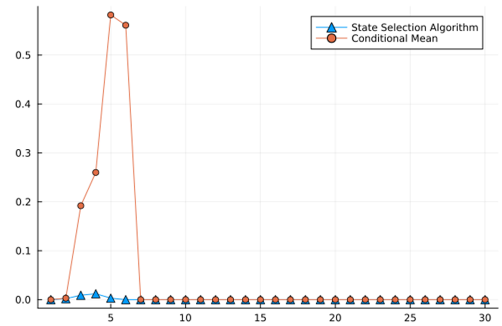

Fig. 4: Comparison of violation rates for initial numerical example

A. *Safety and Performance*

Initially, the State Selection Algorithm was demonstrated on a simple numerical example where a particle density moved around an L-shaped obstacle, as shown in Figure 3, resulting in a drastically lower state constraint violation rate compared to using Certainty Equivalence with the mean state value.

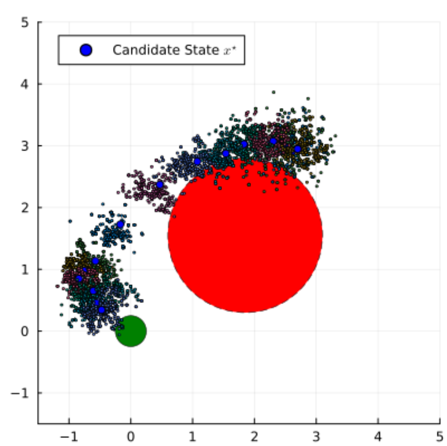

Fig. 5: QP-CBF obstacle avoidance with no SSA

Next, the SSA was applied onto the QP-CBF obstacle avoidance problem. In the displayed figures, the blue dots highlight the mean state value in the case without SSA and the minimum cost feasible state in the case with SSA. In the deterministic, zero process noise case, the QP-CBF safety filter formally guarantees that the mean state will not enter the unsafe set. Since there are perturbations due to process noise, this guarantee no longer applies, but the safety-filter works well to keep the mean state away from the obstacle as much as possible. However, much of the probability distribution besides the mean state value enters the unsafe set, which can be dangerous considering that the distribution represents likely estimates of the true state of the system. This particularly highlights the need for a robust method of state estimation for deterministic safety controllers.

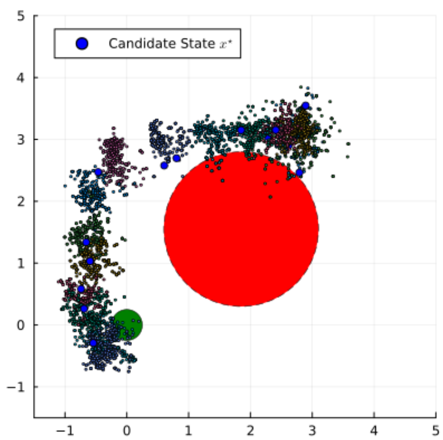

Fig. 6: QP-CBF obstacle avoidance with SSA

Alternatively, the QP-CBF safety-filter combined with the State Selection Algorithm results in a more conservative path around the obstacle. When the particle density is closer to the unsafe set, the candidate state tends to be a state closer to the obstacle, as state estimates close to the obstacle result in more conservative control actions, which lead to lower constraint violation rates. Conversely, as the distribution moves away from the obstacle, the candidate state aligns closely with the mean state because there is more room to optimize over the cost function (3).

Notably, a common problem with control barrier function techniques is the tendency to get stuck when facing a circular obstacle head-on, specifically in the deterministic case. This same issue can be recreated in this problem setup where the particle density has trouble moving around the obstacle if it starts near this CBF "dead-zone". Interestingly, the QP-CBF with SSA can often keep away from the

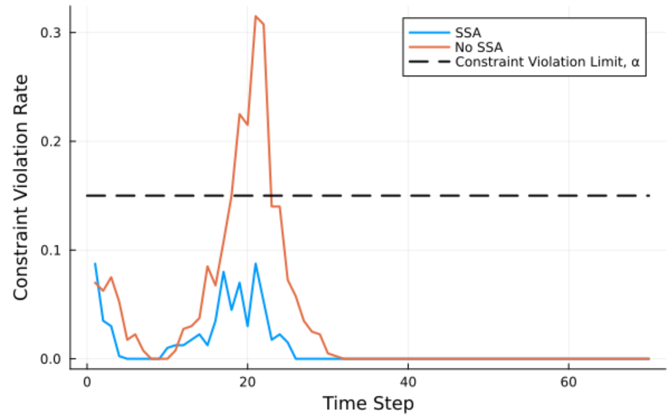

Fig. 7: QP-CBF constraint violation rate comparison

obstacle more effectively than without SSA in this situation.

The running cost of the two scenarios are displayed in Figure 8. In this instance, the cost of SSA is similar to the cost of Certainty Equivalence without SSA. As stated in [3], it should be guaranteed that the cost of the selected state $x^\star$ is less than or equal to the cost of density's sample average as an immediate consequence of the minimization over x_0'. However, this only applies if the sample average is a feasible state. In this obstacle avoidance problem, the mean state estimate is often not a feasible state, as demonstrated with the high state violation rates shown in Figure 5 and Figure 7, so the State Selection Algorithm chooses a higher cost state estimate in order to satisfy safety constraints. Overall, the SSA results in a similar performance measure as quantified by the LQR cost metric, but it results in considerably higher safety.

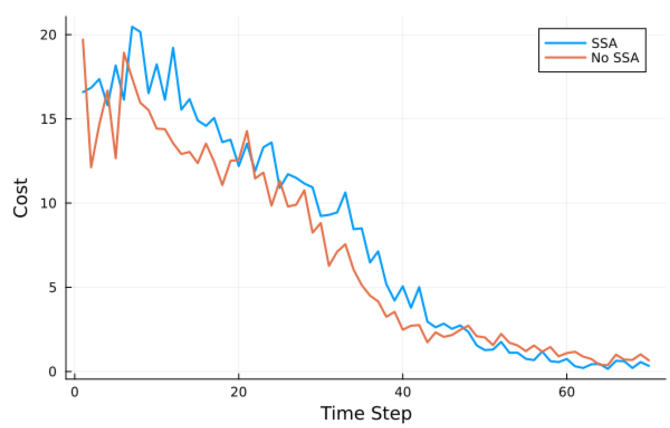

Fig. 8: Running cost comparison

B. *Computation*

This additional measure of safety comes with the downside of a higher computational load. Compared to the mean-based Certainty Equivalence method, the State Selection Algorithm requires the additional calculation of $L*M$ trajectories, each N steps long, which is expected with a nested Monte Carlo structure. Like many Monte Carlo methods, each simulation is completely independent, so theoretically the entire SSA can be completely parallelized with GPU programming and is left as future work.

V. Conclusion

In this project, we proposed and demonstrated a method for robust state estimation to improve the safety of deterministic safe control algorithms within stochastic environments. Leveraging techniques from Monte Carlo and stochastic MPC, we can sample simulated open-loop trajectories associated with the state estimates provided by a particle filter and minimize a cost function over the set of feasible states. This showed improvements to safety in comparison to a nominal mean-state-value-based Certainty Equivalence feedback method when applied to a Quadratic-Programming Control Barrier Function obstacle avoidance problem.

Due to the modular nature of the State Selection Algorithm, various controllers, cost functions, and particle filters can be used to fit different objectives. For instance, there is work being done to use the State Selection Algorithm with an Extended Kalman Filter as the dynamical system where the cost function to be minimized involves the mean and variances, which can result in a better state estimate for nonlinear observation models. Other future work includes using neural network controllers to speed up computation, using an Extended Kalman Filter for the particle filter, tackling higher order control barrier functions, and implementing GPU parallelization.

References

[1] S. Bansal, M. Chen, S. Herbert, C. Tomlin (2017). ”Hamilton-Jacobi Reachability: A Brief Overview and Recent Advances.”

[2] A. Ames, X. Xu, J. Grizzle, P. Tabuada (2017). ”Control Barrier Function Based Quadratic Programs for Safety Critical Systems.”

[3] Y. Bar-Shalom, (1981) “Stochastic dynamic programming: Caution and probing,” IEEE Transactions on Automatic Control, vol. 26, no. 5, pp. 1184–1195

[4] M. Ramadan, R. Bitmead, K. Huang (2023). ”State estimation for control: an approach to stochastic MPC.”

[5] T. Homem-de-Mello, G. Bayraksan (2014). ”Monte Carlo Sampling-Based Methods for Stochastic Optimization.”

[6] A. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, P. Tabuada (2019). ”Control barrier functions: Theory and applications.” ECC