Optimization-based Estimation of Environment Obstacles from Human Demonstration Using Control Lyapunov Function and Control Barrier Functions

Shafagh Keyvanian, HoJin Choi, Minku Kim

Penn University of Pennsylvania

Introduction

Motivation:

- Robots often rely on sensors for obstacle detection.
- Leveraging expert demonstrations without sensors can enhance robot obstacle recognition in cases of obstacle occlusion.
- Enabling recognition of non-physical obstacles such as unsafe or non-favorable regions in the environment.

Contributions:

- Modeling human behavior around obstacles: Learning α and γ using probabilistic model of expert demonstrations
- Learning position and radius of obstacles: x_{obs} and r_{obs} using *all* human demonstrations (including non-expert demos)
- Multiple obstacles estimation

Background:

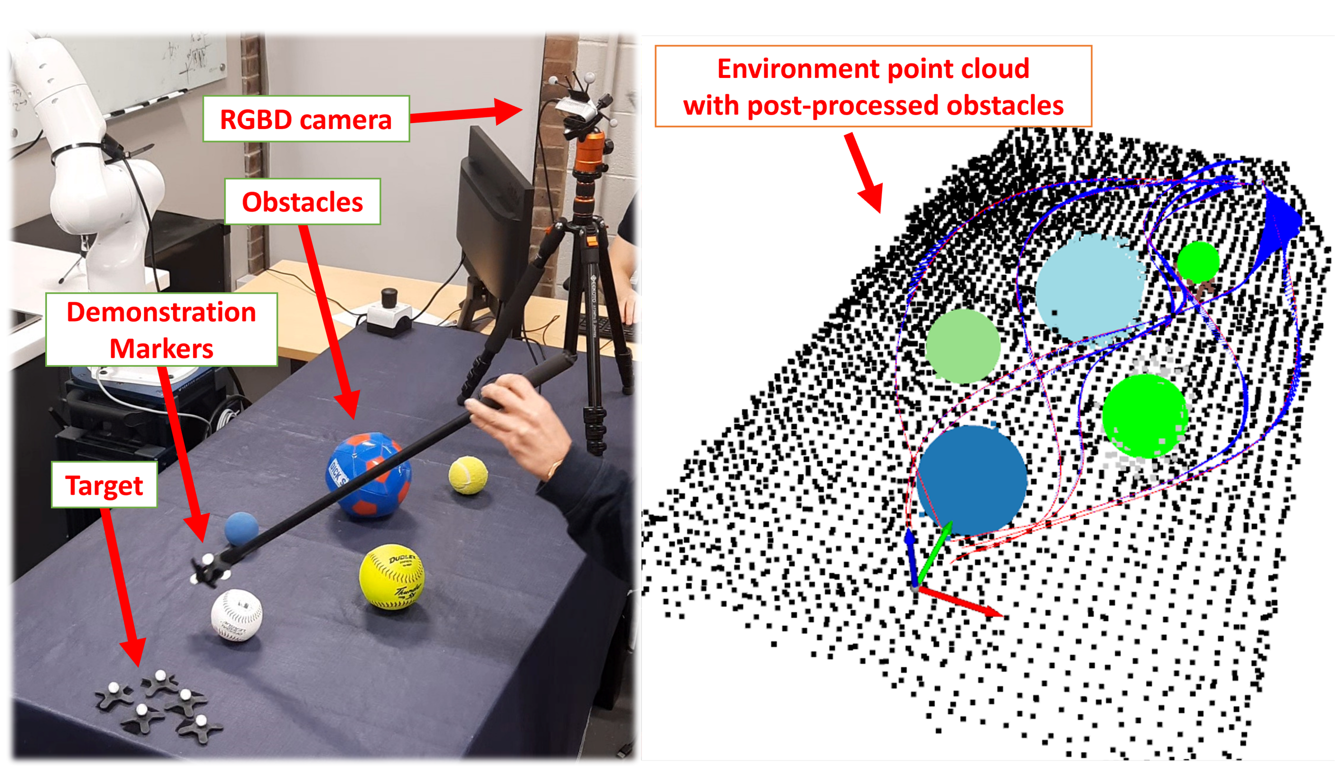
- *Control Lyapunov Function* for stability assurance and *Control Barrier Functions* for safety assurance, as constraints of a QP optimization

$$u(x) = \underset{(u,\delta)\in R^{m+1}}{\arg\min} \ \frac{1}{2}u^T H(x)u + p\delta^2 \qquad \text{(CLF – CBF – QP)}$$

$$\text{s.t.} \quad L_f V(x) + L_g V(x)u \leq -\gamma(V(x)) + \delta$$
$$L_f h(x) + L_g h(x)u \geq -\alpha(h(x))$$

Data Construction

1. **2D optimization-based simulation**
   Trajectory with input constraints and cost for velocity and acceleration using non-linear optimization from Drake
2. **2D human-like demonstrations with mouse**
3. **2D and 3D real human demonstrations**
   Motion Capture system to record demonstrations
   RGBD camera to create point clouds for environments and obstacles



Methods

➤ **Problem Assumptions**

1. Obstacle shapes can be decomposed into multiple spheres.
2. Expert trajectories lead to a stable target without collision.

➤ **Problem Formulation**

$$u_k = x_{k+1} - x_k$$

Lyapunov function: $V(x) = (x_{target} - x)^2$

Barrier function: $h(x) = (x - x_{obstacle})^2 - r^2_{obstacle}$

$\alpha(h(x)) = \alpha h(x)$ and $\gamma(V(x)) = \gamma V(x)$

CLF-CBF-QP:

$$u_{k,\text{estimated}} = \underset{(u,\delta)\in R^{m+1}}{\arg\min} \ ||x_{k+1} - x_k||_2^2 + p\delta^2$$

$$\text{s.t.} \quad 2(x_{target} - x)u \leq -\gamma(x_{target} - x)^2 + \delta$$
$$2(x - x_{obstacle})u \geq -\alpha((x - x_{obstacle})^2 - r^2_{obstacle})$$

Loss function:

$$\text{loss}\ (u_{\text{estimated}}, u_{\text{demo}}) = 1 - \cos\left(u_{\text{estimated}}, u_{\text{demo}}\right)$$

➤ **Step 1: Learning α and γ**

Modeling human-like obstacle avoidance behavior

➜ *GMM-GMR** used to learn human-like trajectory around obstacles from a few expert demonstrations

➜ The resulting regression function used for x_k and u_k

➜ Obstacles known, α and γ unknown

➜ Explore α and γ until Loss value converges.

➤ **Step 2: Obstacle Estimation**

➜ All demonstrations data used for x_k and u_k

➜ α, γ learnt in step 1, obstacles' position ans size unknown

➜ Explore obstacles until Loss value converges

* **GMM-GMR:** Trajectory modeling from expert demonstrations

GMM: Mixture of Gaussian Functions

$$P(x, y) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x, y, \ \mu_k, \Sigma_k)$$

$$\mu_k = \begin{bmatrix}\mu_{x,k} \\ \mu_{y,k}\end{bmatrix}, \Sigma_k = \begin{bmatrix}\Sigma_{x,k} & \Sigma_{xy,k} \\ \Sigma_{yx,k} & \Sigma_{y,k}\end{bmatrix}, \quad \sum_{k=1}^{K}\pi_k = 1$$

GMR: Regression function

$$y(x) = \sum_{k=1}^{K} \beta_k(\mu_{k,y} + \Sigma_{k,yx}(\Sigma_{k,x})^{-1}(x - \mu_{k,x}))$$

$$\beta_k = \beta_k \frac{\pi_k \mathcal{N}(x, \mu_{x,k}, \Sigma_{x,k})}{\Sigma_{i=1}^{K} \pi_i \mathcal{N}(x, \mu_{x,i}, \Sigma_{x,i})}$$

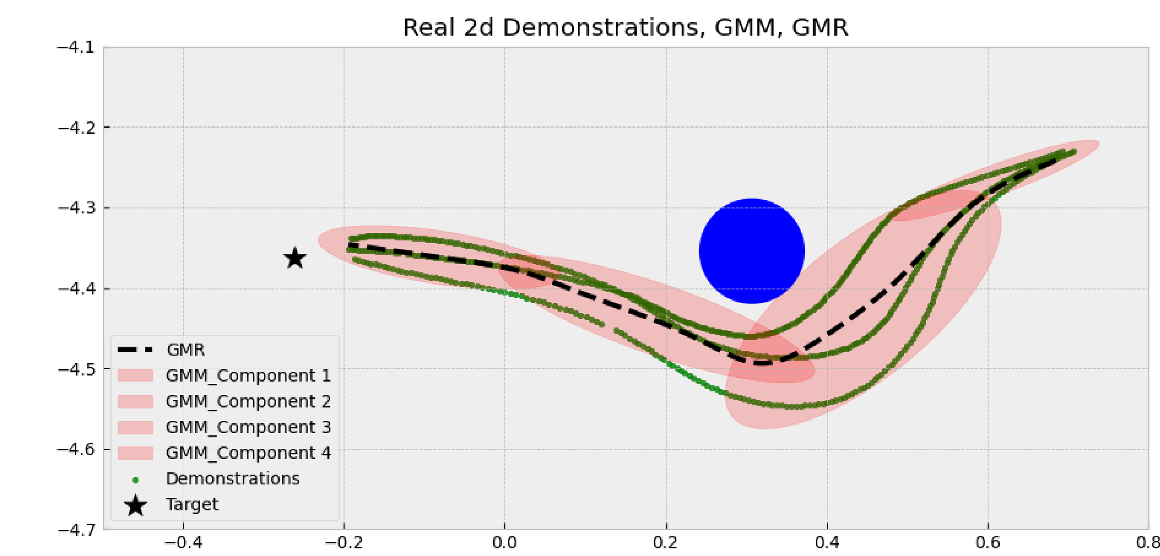
K, π_k, μ_k, Σ_k: # Gaussian kernels, prior probability, mean, covariance matrices

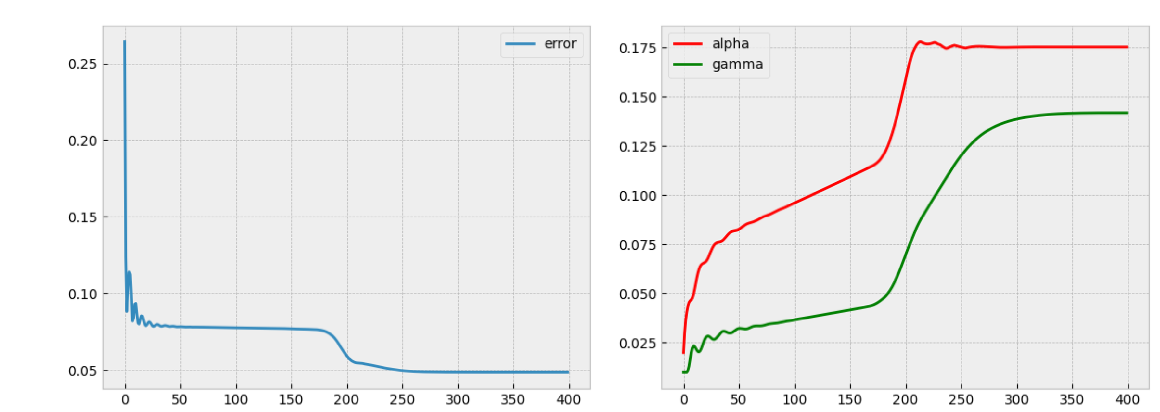
Results

➤ **Step 1: Learning α and γ**

➜ *GMR* Regression function from a few expert demonstrations

➜ Explore α and γ until convergence.





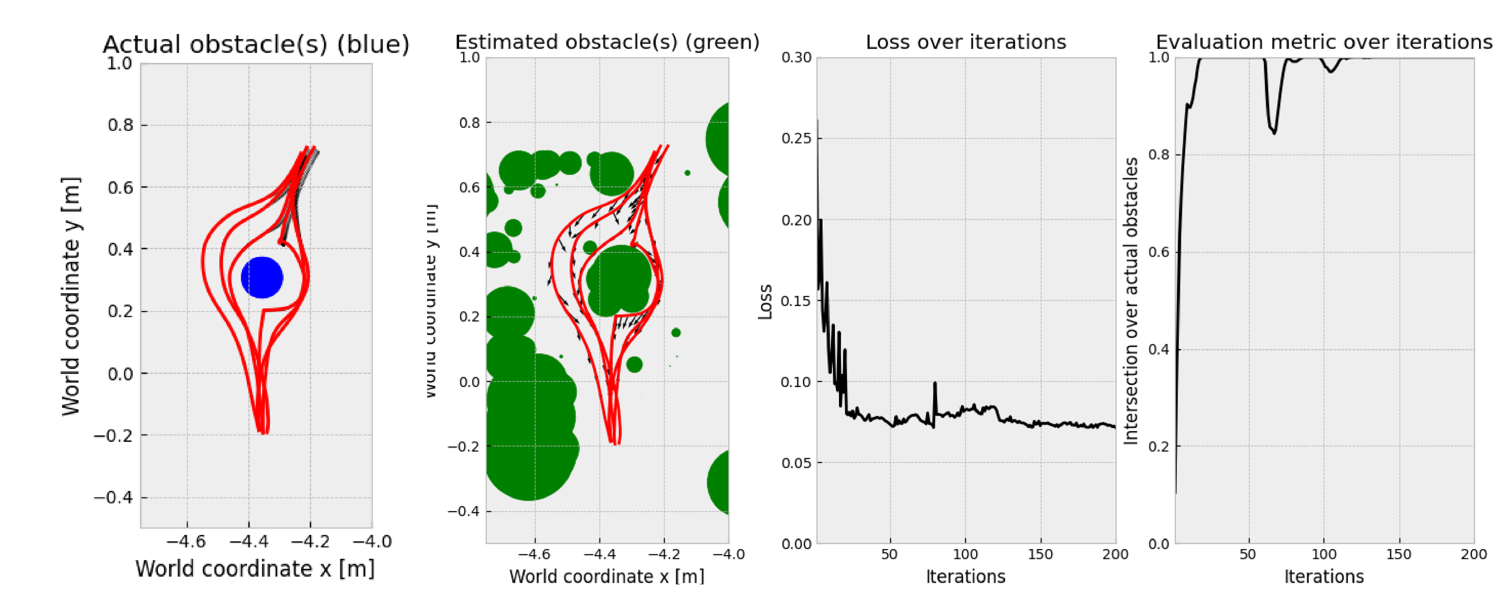
➤ **Step 2: Obstacle Estimation**

➜ All demonstrations data used as x_k and u_k

➜ Explore $x_{obstacles}$ and $r_{obstacles}$ until convergence.

➜ Evaluation Metric: Intersection over actual obstacles $\frac{A \cap B}{A}$ (A: Actual obstacles' area, B: Predicted obstacles' area)

Results for environment with one obstacle:



Results for environment with multiple obstacles: