

# Vision-Based Safe Swimmer Tracking via Robust High-Order Control Barrier Functions

Mohammad Eissa and Wei Xiao

**Abstract**—This paper presents a learning-based end-to-end framework for the safe tracking of a swimmer from image observations to the camera control policy. Specifically, we use a rail-mounted camera with a You-Only-Look-Once (YOLO) model that detects the swimmer and extracts the bounding boxes of the swimmer through camera images. These measurements are then transformed into the state of the swimmer that is further employed to determine the control of the camera’s motion along the longitudinal rail axis so that the swimmer reliably remains inside the pre-determined allowed region of visibility. The reliability of the detection is degraded due to measurement jitter and missed detections which can cause low-confidence images to be used. We address this by quantifying the detection uncertainty offline, and then formulate an online tracking problem using robust High-Order Control Barrier Functions (HOCBFs) to ensure the swimmer safely stays in the field of view of the camera for monitoring purposes. The effectiveness of the proposed framework is demonstrated via simulations conducted in a Unity-based simulator under realistic detection noise.

## I. INTRODUCTION

Using vision-based methods for tracking athletes is becoming an increasingly important tool for sports performance analysis [3]. These methods require extensive manual work, and this challenge is amplified even more in swimming due to the need for dynamic underwater footage [1], [4], [5]. A camera that is able to follow a swimmer autonomously is a useful tool allowing coaches and swimmers to get important swimming metrics in an easier and faster way. The challenge with underwater imagery arises from reflections, bubbles, and occlusions that can alter a camera’s view [6]–[8]. As a result, image-based swimmer detection is often unreliable due to many uncertainties.

The use of marker-less camera-based systems have shown promise in recent works. For example, Giulietti et al. [1] introduced SwimmerNET, which uses an underwater 2D swimmer pose-estimation framework that allows coaches to evaluate performance. That result supports the broader consensus that the use of computer vision for analysis without any reliance on wearable sensors or optical markers is possible. However, in order to do this, it is assumed that the swimmer must remain sufficiently visible in all recorded frames.

A more recent full-lane butterfly analysis system by Chern et al. [2] further displays that side-view swimmer bounding boxes are sufficient to attain important swimming metrics like split times and kick-angle analysis. Past work has

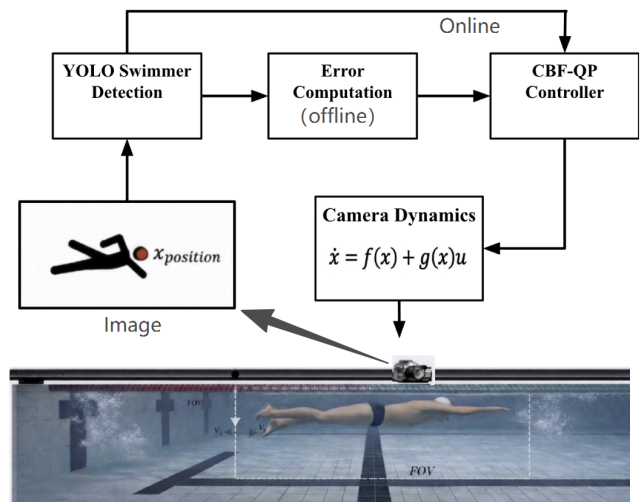


Fig. 1. System architecture of the proposed framework. The camera takes images for the YOLO swimmer detection and real-time state estimations. To account for uncertainties in perception, we evaluate errors offline. Both the online state estimator and offline uncertainty quantifier are used for the robust CBF-QP safe filter to control the camera such that the swimmer safely stays within its field of view.

focused only on performance with the assumption that the images are readily available. In practice, we need to consider how to obtain the images in a flexible and low-cost manner such as prioritizing a safe, closed loop camera control that does not require a multi-camera setup. In this case, visibility is not only used for performance tracking, but also as a real-time safety requirement as the detector output is used directly inside the feedback loop of a moving camera that usually has limited field of view.

Since we use a feedback loop for perception outputs, any uncertainty can cause safety violations due to loss of visibility. Thus, the swimmer must remain inside a prescribed image-space region. If the controller reacts too aggressively to noisy detections, or fails to account for missed detections and uncertainty, the swimmer can leave the admissible region and tracking quality degrades. This motivates a learning-based safe and robust controller for the camera to ensure the reliable monitoring of the swimmer.

In the literature, Control Barrier Functions (CBFs) provide a structured mechanism to enforce these constraints in safety-critical tracking [17], [18], [20]. The rail-mounted camera is modeled as a high-order system and image-space visibility constraints depend on camera position through the projection map. This motivates a high-order CBF (HOCBF) formulation

M. Eissa is with the Data Science Program, Worcester Polytechnic Institute, Worcester, MA, USA (e-mail: meissa@wpi.edu).

W. Xiao is with WPI and MIT, MA, USA (e-mail: weixy@mit.edu).

[19], [21], [22]. Since measurements are perception-driven, uncertainty must be incorporated explicitly in the safety filter [23], [24].

The focus of this paper is on the safety portion of the swimmer-tracking pipeline. Through YOLO-based perception [9], confidence-aware measurement conditioning, uncertainty-aware state estimation, and robust HOCBF-based safety filtering, this paper proposes a framework that leads to a controller guaranteeing that the swimmer remains within the prescribed safety constraints, i.e., the field of view of the camera, under the assumed bounded uncertainty model. The full system, illustrated in Fig. 1, operates as follows. A moving pool-side camera sends rendered frames to a YOLO detector. The detector outputs swimmer bounding boxes and confidence scores, which are filtered and converted into image-space measurements. These measurements are then fed into a robust CBF-based controller for the camera that tracks the swimmer while ensuring that the swimmer remains inside a prescribed safe region of the frame and that the camera respects rail and actuator limits.

The main contributions of this paper are as follows:

- We propose a learning-based uncertainty-aware tracking architecture for swimmer monitoring using a rail-mounted camera and YOLO-based swimmer detection.
- We employ a robust HOCBF formulation to ensure the visibility (i.e., safe tracking) of a swimmer via offline uncertainty quantification and online state estimations under the assumed bounded uncertainty model.
- The effectiveness of the framework is demonstrated in a Unity-based simulator to evaluate safe swimmer tracking under detection noise, missed measurements, and dynamic swimmer motion.

## II. PROBLEM FORMULATION

Consider a camera mounted on a rail parallel to a swimming lane as shown in Fig. 1, where  $z_c(t) \in \mathbb{R}$  denotes the camera position along the rail,  $v_c(t) \in \mathbb{R}$  denotes its longitudinal velocity, and  $u(t) \in \mathbb{R}$  denotes the commanded longitudinal acceleration (control input).

The camera dynamics are given as

$$\dot{x} = f(x) + g(x)u, \quad (1)$$

where  $x = [z_c \ v_c]^T$ ,  $f : \mathbb{R}^2 \rightarrow \mathbb{R}^2$  and  $g : \mathbb{R}^2 \rightarrow \mathbb{R}^2$  are locally Lipschitz continuous.

**Safe tracking constraint.** Let  $z_s(t)$  denote the swimmer's longitudinal position. Because we use YOLO, the controller does not directly measure  $z_s(t)$ . Instead, the perception system returns an image-space bounding box  $I(t) = [\hat{x}_{\min}, \hat{x}_{\max}, \hat{y}_{\min}, \hat{y}_{\max}]$  with confidence score  $c(t) \in [0, 1]$  [15], [16]. The coordinates of the swimmer box edges are given by projection maps  $x_{\min}(I(t)) = \phi_{\min}(I(t), S_s)$  and  $x_{\max}(I(t)) = \phi_{\max}(I(t), S_s)$ , where  $\phi_{\min}, \phi_{\max}$  are two mapping functions from the swimmer bounding box to the swimmer location and  $S_s$  denotes the swimmer body extent. The safe tracking constraint (or the visibility constraint) is to maintain the swimmer inside the prescribed image-space:

$$x_L(z_c) \leq x_{\min}(I), \quad x_{\max}(I) \leq x_R(z_c), \quad (2)$$

where  $[x_L(z_c), x_R(z_c)]$  denotes the admissible horizontal interval of the field of view box of the camera. We may define a safe set (visible space)  $\mathcal{C}_x = \{z_c \in \mathbb{R} : x_L(z_c) \leq x_{\min}(I), x_{\max}(I) \leq x_R(z_c)\}$  based on the above constraint.

The mapping we use is generally non-linear, but around the current camera position it admits local first- and second-order derivatives with respect to  $z_c$ . These derivatives are used by the controller to construct HOCBFs.

**Rail safety constraint.** The rail is bounded, so the camera must satisfy

$$z_{\min} \leq z_c(t) \leq z_{\max}, \quad (3)$$

where  $z_{\min} \in \mathbb{R}, z_{\max} \in \mathbb{R}$  denote the limits of the swimming pool.

**State limitation and control bound.** The camera system has to satisfy a speed constraint and a control bound:

$$|v_c(t)| \leq v_{\max}, \quad |u(t)| \leq u_{\max}. \quad (4)$$

**Objectives.** The control design has two coupled objectives. First, we hope to keep the swimmer centered in the field of view of the camera for a better full-body analysis. Second, we hope to minimize the control input effort in the form of

$$\min_u \int_0^T u^2 dt \quad (5)$$

to save energy and to make the control input remain smooth so that the motion is always stable visually, where  $T$  is the final time. If a nominal controller  $u_{\text{nom}}$  is available, we may minimize deviation from  $u_{\text{nom}}$  in the above cost function.

**Problem 1:** Given the image observations of the camera, find a safety-filtered controller for camera system (1) that seeks to minimize (5) and stabilizes the swimmer to the center of the field of view of the camera, while satisfying all the safety constraints (2) (3) and camera system limitation and control bound (4).

**Problem solution.** Given the camera images, we employ a YOLO model to detect the bounding box of the swimmer, and then map the bounding box to the swimmer left and right position limits  $x_{\min}, x_{\max}$ . We further evaluate the uncertainties of  $x_{\min}, x_{\max}$  offline by comparing the results from the YOLO model and the ground truth, and obtain the uncertainty bound. During inference time, the real time state estimations from the YOLO model and the uncertainty bound are then used to construct a robust HOCBF constraint to strictly enforce the safe tracking constraint (2). All the other constraints are implemented by CBFs/HOCBFs, and we can finally get a CBF-based QP controller as in [17], [18], [20] for controlling the camera.

**Assumptions.** We assume a single swimmer in the lane at the center of the pool, the camera motion is constrained to the longitudinal rail axis with fixed yaw, and we have bounded perception uncertainty obtained from offline YOLO-versus-ground-truth analysis. We also assume that all detector outputs are fresh and explicitly marked as stable within a bounded time window. Outdated measurements are discarded to reduce jitter.

### III. SYSTEM ARCHITECTURE AND IMAGE-BASED PERCEPTION

#### A. Closed-Loop Architecture

The closed-loop operation proceeds as follows. The rail-mounted camera captures frames and transmits them to a YOLO detector, which outputs swimmer bounding boxes and confidence scores. These outputs are then filtered and sanity-checked to produce control-relevant image-space measurements, such as the coordinates of the bounding box of the swimmer in the cartesian space. A nominal tracking controller computes a desired control input  $u_{\text{nom}}$  that keeps the swimmer at the center of the field of view of the camera, and a robust HOCBF-QP safety filter modifies this input only when required to satisfy rail and visibility constraints. In this architecture, the detector also provides uncertainty quantification offline, while safety-critical enforcement (i.e., the control of the camera) is performed by the HOCBF-QP layer.

#### B. YOLO Measurement Conditioning

The perception pipeline is based on the YOLO family [9]–[12], trained for underwater swimmer detection. Raw outputs are not passed directly to control; instead, they are filtered before entering the feedback loop by the following techniques:

**Confidence gating:** Detections with low confidence are rejected. This prevents low-quality boxes from producing aggressive control actions.

**Temporal association:** When multiple candidate boxes are present, the detector output is associated with the previously predicted swimmer location. This reduces identity swaps and abrupt jumps.

**Filtering:** Accepted image-space measurements are filtered using a constant-velocity Kalman filter with innovation gating before being used for control [13], [14]. This stage reduces jitter and rejects outlier detections while preserving responsiveness to genuine swimmer motion.

**Short-dropout handling:** When fresh detections are unavailable for a brief period, the system either holds the last reliable box or predicts it forward for a short horizon. This avoids unnecessary switching between aggressive catch-up and full-stop behavior.

On the Unity side, incoming boxes are also checked for physically implausible geometry or frame-to-frame jumps. Boxes with unrealistic area, position jump, or size change are discarded before being used by the controller.

#### C. Confidence-Aware Measurement

The online controller uses a confidence-aware uncertainty estimate rather than a fixed detection variance. Let  $\hat{x}_k$  denote the accepted image-space measurement at time step  $k$ , and let  $\bar{x}_k$  denote its filtered value.

The residual  $r_k = \hat{x}_k - \bar{x}_{k-1}$  drives an exponentially smoothed variance estimate:

$$\sigma_{x,k}^2 = (1 - \lambda) \sigma_{x,k-1}^2 + \lambda r_k^2, \quad (6)$$

where  $\sigma_{x,k}^2$  is the exponentially weighted variance of the image-space measurement at step  $k$ , and  $\lambda \in (0, 1)$  is a smoothing parameter that controls the adaptation rate.

Lower confidence and larger residuals therefore increase the uncertainty estimate used by the controller.

The perception pipeline thus produces two distinct uncertainty signals, each serving a different role in the controller. The *Kalman filter with innovation gating* described above provides the smoothed state estimate  $\bar{x}_k$  that is used as the measurement input to both the nominal tracking controller and the HOCBF safety filter. The *residual-based Exponential Moving Average (EMA) variance*  $\sigma_{x,k}^2$  is an online measure of instantaneous detection reliability; it is used to contract the image-space centering target region when confidence is low and to enlarge the dynamic safety margin  $m_{\text{dyn}}$  in the visibility tightening of Section IV-D. Neither of these online quantities replaces the offline robust bounds  $\varepsilon_h, \varepsilon_{\dot{h}}, \varepsilon_{\ddot{h}}$  described next, which provide the worst-case uncertainty set for the robust HOCBF constraints.

#### D. Offline Uncertainty Identification

To enable the robust HOCBF safety filter, YOLO outputs are quantified offline with ground-truth image-space trajectories generated in simulation. The resulting error signals include position, velocity, and acceleration discrepancies of the swimmer detection between the perception-derived and ground-truth image-space quantities.

Since the robust HOCBF condition must hold for *all possible* states in the uncertainty set, the offline bounds must cover the full range of observed perception error rather than a statistical fraction of it. We therefore use the maximum absolute error over all offline evaluation sequences as the robust margins  $E$  to quantify the uncertainty of  $I, \dot{I}, \ddot{I}$  for  $x_{\min}(I), x_{\max}(I)$ , expressed in normalized image-space units and their time derivatives. Although  $I$  is four-dimensional, the controller-side uncertainty model is parameterized by a single scalar horizontal image coordinate (edge-gap state) along the rail-tracking axis, while residual box-width and pose-induced variations are absorbed into the bounded uncertainty set  $E$ . The specific values used in our implementation are reported in the simulation.

Using maximum rather than percentile-based bounds is a deliberate design choice: because a single constraint violation can cause the swimmer to leave the admissible region, the safety filter must guard against the worst observed error, not merely a likely one. The cost of this conservativeness is a modest reduction in nominal tracking bandwidth, since the feasible acceleration interval is tightened by the worst-case margins. In practice, the resulting performance loss is small because the QP can still track the swimmer accurately within the tightened interval, as confirmed by the centering-error statistics in the simulation section.

### IV. ROBUST HOCBF-QP CONTROLLER DESIGN

#### A. Nominal Tracking Controller

The nominal controller is designed to track swimmer motion smoothly before safety constraints are enforced. Let

$z_d(t)$  and  $v_d(t)$  denote the desired rail position and velocity inferred from the swimmer estimate.

The nominal acceleration control for the camera is defined as

$$u_{\text{nom}} = k_p(z_d - z_c) + k_d(v_d - v_c) + u_{\text{center}} + u_{\text{damp}} \quad (7)$$

where  $u_{\text{damp}} = -k_{\text{damp}} \cdot \gamma(v_d) \cdot v_c$  ( $k_{\text{damp}} > 0$ ) is a time-varying low-speed damping term and  $\gamma(v_d) \in [0, 1]$  is a scaling factor that increases as the desired speed  $|v_d|$  decreases, so that damping is strongest when the camera is near-stationary or changing direction and inactive at higher speeds, and  $u_{\text{center}}$  is an image-space centering term. For centering, the error  $d = x_s - x_d$  between the current swimmer image coordinate  $x_s$  and the desired center  $x_d$  drives a desired image-space velocity  $\dot{x}_d = -k_x d$ ,  $k_x > 0$ , which is mapped to a camera velocity via the local projection slope  $f_1 = \partial x_s / \partial z_c$  as  $v_{\text{center},d} = \dot{x}_d / f_1$ , giving  $u_{\text{center}} = k_v(v_{\text{center},d} - v_c)$ ,  $k_v > 0$ .

The nominal acceleration is then smoothed via exponential moving average filtering and clamped by a jerk limit  $|\dot{u}_{\text{nom}}| \leq \dot{j}_{\text{max}}$  before being passed to the safety filter.

### B. Why High-Order Barrier Functions Are Required

The control input  $u$  enters the dynamics at the acceleration level, not at velocity. The visibility constraints depend on  $z_c$  through the camera projection map, so the input does not appear in the first time derivative of the visibility barrier. This makes the visibility safety constraint have relative degree two with respect to  $u$ , and a standard first-order CBF is therefore not sufficient [19], [21], [22].

For a scalar barrier function  $b(x)$  of relative degree two with respect to the dynamics (1), the first derivative is independent of the input,

$$\dot{b}(x) = L_f b(x), \quad (8)$$

and the second derivative is affine in  $u$ ,

$$\ddot{b}(x) = L_f^2 b(x) + L_g L_f b(x) u. \quad (9)$$

Using the exponential-HOCBF construction [22], define  $\psi_0(x) = b(x)$  and

$$\psi_1(x) = L_f b(x) + \alpha_0 b(x), \quad (10)$$

with  $\alpha_0 > 0$ . The HOCBF condition  $\dot{\psi}_1(x) + \alpha_1 \psi_1(x) \geq 0$  with  $\alpha_1 > 0$  expands to

$$L_f^2 b(x) + L_g L_f b(x) u + (\alpha_0 + \alpha_1) L_f b(x) + \alpha_0 \alpha_1 b(x) \geq 0. \quad (11)$$

Equation (11) is a second-order CBF used throughout the controller design. The system is safe whenever  $b(x) \geq 0$ ; a *safety violation* occurs when  $b(x) < 0$ , meaning the swimmer has left the admissible region. Since (11) is affine in  $u$ , it can be enforced directly as a linear inequality in the CBF-QP.

### C. Rail Safety Constraints

The lower and upper rail barrier functions are

$$b_{\min}(x) = z_c - z_{\min}, \quad b_{\max}(x) = z_{\max} - z_c. \quad (12)$$

Since  $L_f b_{\min}(x) = v_c$ ,  $L_f^2 b_{\min}(x) = 0$ , and  $L_g L_f b_{\min}(x) = 1$ , the HOCBF condition (11) yields

$$u + (\alpha_{r0} + \alpha_{r1})v_c + \alpha_{r0}\alpha_{r1}(z_c - z_{\min}) \geq 0. \quad (13)$$

Analogously,  $L_f b_{\max}(x) = -v_c$  and  $L_g L_f b_{\max}(x) = -1$  give

$$-u - (\alpha_{r0} + \alpha_{r1})v_c + \alpha_{r0}\alpha_{r1}(z_{\max} - z_c) \geq 0. \quad (14)$$

Near a rail endpoint, higher speed requires stronger braking, which the above two HOCBF constraints enforce directly. We also use a CBF to enforce the speed limitation of the camera in (4).

### D. Safe Tracking / Two-Sided Visibility Constraints

To keep the entire swimmer inside the safe region, we define the left and right box-edge barrier functions

$$b_L(x) = x_{\min}(I) - x_L(z_c), \quad b_R(x) = x_R(z_c) - x_{\max}(I). \quad (15)$$

Since  $x_{\min}(I)$ ,  $x_{\max}(I)$  depends on the location of the swimmer from the YOLO model, they are also time-varying. Therefore, we need to consider them in the HOCBF constraint. Since the safety constraints have relative degree two with respect to  $u$ , the left-edge HOCBF constraint is then given by:

$$\begin{aligned} & L_g L_f b_L(x) u + L_f^2 b_L(x) + \frac{\partial b_L(x)}{\partial x_{\min}(I)} \ddot{x}_{\min}(I) \\ & + (\alpha_0 + \alpha_1) (L_f b_L(x) + \frac{\partial b_L(x)}{\partial x_{\min}(I)} \dot{x}_{\min}(I)) + \alpha_0 \alpha_1 b_L(x) \geq 0. \end{aligned} \quad (16)$$

In the above,  $x_{\min}(I)$ ,  $\dot{x}_{\min}(I)$ ,  $\ddot{x}_{\min}(I)$  are from the YOLO model; they explicitly depend on  $I, \dot{I}, \ddot{I}$  and are therefore subject to uncertainties.

The right-edge HOCBF constraint is given by

$$\begin{aligned} & L_g L_f b_R(x) u + L_f^2 b_R(x) + \frac{\partial b_R(x)}{\partial x_{\max}(I)} \ddot{x}_{\max}(I) \\ & + (\alpha_0 + \alpha_1) (L_f b_R(x) + \frac{\partial b_R(x)}{\partial x_{\max}(I)} \dot{x}_{\max}(I)) + \alpha_0 \alpha_1 b_R(x) \geq 0. \end{aligned} \quad (17)$$

The terms  $x_{\max}$ ,  $\dot{x}_{\max}$ ,  $\ddot{x}_{\max}$  also explicitly depend on  $I, \dot{I}, \ddot{I}$  and are therefore subject to uncertainties.

Equations (16) and (17) are the nominal visibility HOCBF constraints. Their robust counterparts are obtained by replacing  $x_{\min}$ ,  $\dot{x}_{\min}$ ,  $\ddot{x}_{\min}$  or  $x_{\max}$ ,  $\dot{x}_{\max}$ ,  $\ddot{x}_{\max}$  with their worst-case evaluations over the uncertainty set built from the bound  $E$  identified in Section III-D.

In the implementation, the required derivatives are obtained either from local finite differences of the Unity projection map (geometric mode) or from YOLO-derived image-space edge dynamics combined with an online linear  $x$ - $z$  mapping (perception-driven mode).

**Dynamic Safe-Set Tightening.** Purely static bounds are often not sufficient when the swimmer box approaches the camera field of view boundary with non-negligible velocity. To trigger earlier intervention, we tighten the left and right admissible bounds with a predictive dynamic margin:

$$m_{\text{dyn}} = \min \{ \bar{m}, |\dot{e}|T_p + 0.5|\ddot{e}|T_a^2 \} \quad (18)$$

where  $\dot{e}$  and  $\ddot{e}$  are edge-gap velocity and acceleration estimates,  $T_p$  and  $T_a$  are prediction horizons, and  $\bar{m} > 0$  is the maximum tightening level. The effective visibility bounds then become  $x_{\text{eff},L} = x_L(z_c) + m_{\text{dyn}}$  and  $x_{\text{eff},R} = x_R(z_c) - m_{\text{dyn}}$ , triggering earlier intervention when the swimmer is moving toward the edge of the frame.  $x_{\text{eff},L}$  and  $x_{\text{eff},R}$  are then used to replace the  $x_L(z_c), x_R(z_c)$  in (15), respectively.

#### E. Robust HOCBFs for Safe Tracking

In this subsection, we consider how to employ the offline uncertainty quantification in the CBF method to build robust HOCBFs for safe tracking.

Let  $\mathbf{I} = (I, \dot{I}, \ddot{I})$ . We can define the following set for  $\mathbf{I}$  with the uncertainty bound  $E$  from the state estimation of the YOLO model:

$$S(E, \mathbf{I}) := \{ \mathbf{y} : \mathbf{I} - E \leq \mathbf{y} \leq \mathbf{I} + E \} \quad (19)$$

Defining  $A_L(x) := L_g L_f b_L(x)$  and  $B_L(x, \mathbf{I}) := L_f^2 b_L(x) + \frac{\partial b_L(x)}{\partial x_{\min}(I)} \ddot{x}_{\min}(I) + (\alpha_0 + \alpha_1)(L_f b_L(x) + \frac{\partial b_L(x)}{\partial x_{\min}(I)} \dot{x}_{\min}(I)) + \alpha_0 \alpha_1 b_L(x)$ , The robust form of the HOCBF constraint (16) becomes

$$\min_{\mathbf{y} \in S(E, \mathbf{I})} [A_L(x)u + B_L(x, \mathbf{y})] \geq 0. \quad (20)$$

where  $B_L(x, \mathbf{y})$  is the same as  $B_L(x, \mathbf{I})$  by replacing  $\mathbf{I}$  with  $\mathbf{y} \in S(E, \mathbf{I})$ .

Similarly, we can get the robust HOCBF constraint for the right-edge visibility constraint:

$$\min_{\mathbf{y} \in S(E, \mathbf{I})} [A_R(x)u + B_R(x, \mathbf{y})] \geq 0, \quad (21)$$

where  $A_R(x) := L_g L_f b_R(x)$  and  $B_R(x, \mathbf{I}) := L_f^2 b_R(x) + \frac{\partial b_R(x)}{\partial x_{\max}(I)} \ddot{x}_{\max}(I) + (\alpha_0 + \alpha_1)(L_f b_R(x) + \frac{\partial b_R(x)}{\partial x_{\max}(I)} \dot{x}_{\max}(I)) + \alpha_0 \alpha_1 b_R(x)$ . In implementation, robustification is applied by additive worst-case margin tightening on the drift term; additionally, sign-aware interval selection for  $L_g L_f b$  is used only when interval uncertainty on this coefficient is explicitly enabled (e.g., to account for mapping-gain uncertainty), otherwise  $L_g L_f b$  is treated as deterministic. The resulting constraints remain affine in  $u$ , so the online solve remains a one-dimensional interval projection.

#### F. Robust CBF-QP Safety Filter

At each control step, the nominal acceleration (control) is projected onto the set of admissible accelerations defined by all active safety constraints [25]. Combining all the previous

---

#### Algorithm 1: Safe swimmer tracking

---

- 1: **Input:** A pre-trained YOLO model, image observations
  - 2: Collect data from the simulator
  - 3: Fine-tune the YOLO model with the data
  - 4: Uncertainty quantification in swimmer state estimation with YOLO and get  $E$
  - 5: Construct CBFs and robust HOCBFs as in (13), (14), (20), (21)
  - 6: Formulate the robust CBF QP safe filter (22)
  - 7: Initialize the camera system
  - 8: **for**  $t \leq T$  **do**
  - 9:   Get the current camera image and camera state  $x$
  - 10:   Estimate the current bounding box of the swimmer using YOLO and get  $\mathbf{I}$ .
  - 11:   Compute nominal control  $u_{\text{nom}}$  as in (7)
  - 12:   Solve the CBF-QP (22) and get  $u^*$
  - 13:   Apply  $u^*$  to control the camera and update the dynamics (1)
  - 14: **end for**
  - 15: **return** A safe tracking trajectory for the swimmer
- 

subsections, we can get the following safety filter:

$$\begin{aligned} u^* &= \arg \min_u \frac{1}{2}(u - u_{\text{nom}})^2 \\ \text{s.t. } & (13), (14), (20), (21) \\ & \text{CBF for speed limitation,} \\ & |u| \leq u_{\text{max}}. \end{aligned} \quad (22)$$

Before implementing the proposed framework online, we first fine-tune the YOLO model with the data collected from the simulator and evaluate the uncertainty of the swimmer state offline and obtain  $E$ . During online implementation, each control cycle begins with a perception stage: the controller receives the latest YOLO detection packet and confidence score, rejects implausible or low-confidence measurements, and updates the filtered swimmer state and uncertainty estimates. A nominal tracking acceleration  $u_{\text{nom}}$  is further computed from the filtered state. Then, we formulate CBFs or robust HOCBFs for all the safety constraints. Finally, we solve the CBF-QP (22) and obtain the filtered control  $u^*$  for the camera. This process is repeated until the final time. The complete procedure is given in Algorithm 1.

## V. SIMULATIONS AND RESULTS

### A. Simulation Setup

**Unity Environment.** The framework that we propose here is evaluated in a Unity simulation of a swimmer moving along an Olympic-sized swimming pool (50 meters) while a rail-mounted camera tracks the swimmer from the side wall. The camera is modeled as double integrator dynamics with bounded velocity and control input.

In the current implementation, the controller discretization step is fixed at 0.01 s (100 Hz), while the target frame rate is 120 Hz for rendering and runtime pacing. The representative limits are chosen as  $v_{\text{max}} = 2.3$  m/s and  $u_{\text{max}} = 20$  m/s<sup>2</sup>.

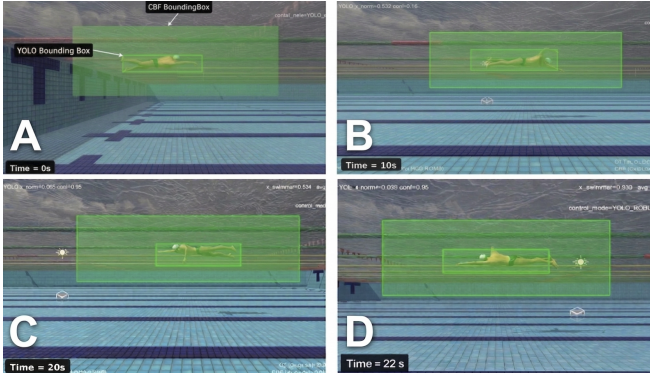


Fig. 2. Unity runtime snapshots of the rail-camera tracking experiment, showing swimmer visibility, YOLO overlays, and safety-box behavior.

The value of  $v_{\max}$  is chosen to exceed typical competitive swimmer speeds, while  $u_{\max}$  permits rapid but bounded camera readjustment when large tracking corrections are required. All reported runtime measurements were collected on an Apple MacBook Pro with an Apple M4 chip and 24 GB RAM.

As shown in Fig. 2, the camera maintains a fixed inward-facing orientation and moves only along the longitudinal rail axis. This isolates the longitudinal tracking problem and matches the current control design. The swimmer motion can be either constant-lap or dynamically varying, which allows the same framework to test nominal tracking, turn transitions, and disturbances in the swimmer speed profile.

**Detector Training and Inference Setup.** The YOLO detector is further trained (fine-tuned) for a single swimmer class using a dataset with 1428 training images, 102 validation images, and 102 test images. The current training configuration uses  $640 \times 640$  inputs for 20 epochs. On the validation split, the detector achieves mAP@50 of 96.3%, precision of 92.5%, and recall of 92.5%. Here, mAP@50 (mean Average Precision at 50% Intersection-over-Union threshold) measures the average detection accuracy across confidence thresholds, precision is the fraction of detections that are correct, and recall is the fraction of ground-truth instances that are detected.

Inference is performed externally and communicated back to Unity through UDP packets containing the horizontal estimate, confidence value, and bounding-box coordinates.

The two-process architecture separates rendering/control from perception inference, which simplifies synchronization and allows packet freshness and asynchronous sensing effects to be handled explicitly at the control layer.

**Controller and YOLO Parameterization.** The controller uses separate HOCBF gain pairs for rail safety and safe tracking constraint (image-space visibility). The nominal command is smoothed and jerk-limited to reduce stop-go behavior under noisy detections. The visibility safe set is defined by a fixed image-space box (determined by the field of view of the camera) with an additional inner margin and dynamic edge tightening based on estimated edge motion. Static tightening captures uncertainty, while dynamic tight-

ening anticipates imminent boundary approach.

We parameterize the swimmer bounding box from the YOLO model with a single scalar horizontal image coordinate aligned with the rail-tracking axis, and treat residual box-width/pose variations as bounded uncertainty captured by  $E$ . The robust margins obtained from the offline YOLO-versus-ground-truth analysis described in Section III-D are  $\varepsilon_h = 0.399$ ,  $\varepsilon_{\dot{h}} = 0.724$ , and  $\varepsilon_{\ddot{h}} = 2.54$  (i.e.,  $E = (\varepsilon_h, \varepsilon_{\dot{h}}, \varepsilon_{\ddot{h}})$  after we parameterize the swimmer bounding box with a single parameter), expressed in normalized image-space units and their time derivatives.

**Evaluation Metrics and Comparisons.** The evaluation emphasizes safety, tracking quality, and control effort. The main metrics are the minimum visibility margin (from the safe tracking constraint), the percentage of frames satisfying image-space safety constraints, swimmer centering error statistics, rail tracking error, mean and peak control magnitude  $|u^*|$ , and the frequency of HOCBF activations.

To isolate the contribution of robust perception-aware safety filtering, we compare nominal tracking without safety filtering, non-robust HOCBF filtering, and robust HOCBF filtering with maximum-error uncertainty bounds under constant-speed and dynamically varying swimmer motion.

## B. Results and Discussion

**Perception Error Characterization.** We first evaluate perception quality by comparing YOLO-derived image-space states against ground truth over time. Fig. 3 summarizes the mean error behavior for position, velocity, and acceleration channels used by the controller. These error trends motivate the robust margins introduced in Section III-D and provide the basis for uncertainty-aware safety filtering.

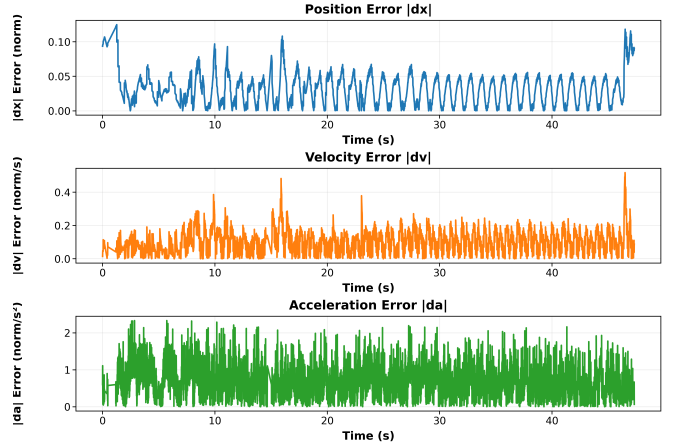


Fig. 3. Per-channel absolute error subplots for position, velocity, and acceleration channels over time between the ground truth state of the swimmer and the YOLO-based detector from a singular run.

**Safety Evaluation.** Safety is evaluated directly through the CBF profile and violation statistics. Fig. 4 reports the CBF profile  $b(x)$  over time for the robust HOCBF-QP controller under YOLO-based sensing with dynamically varying swimmer motion, with the baseline  $b(x) = 0$  shown in red. The CBF profile remains nonnegative throughout the run,

indicating zero visibility constraint violations and confirming that the swimmer stays inside the admissible image-space safe set for all times. However, the non-robust HOCBF may violate the safety constraint due to YOLO detection uncertainties.

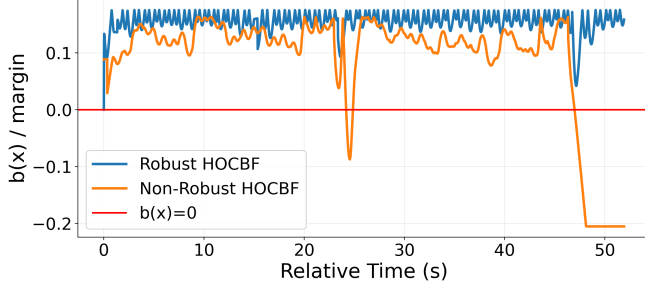


Fig. 4. CBF profile  $b(x)$  over time corresponding to the safe tracking constraint for the robust HOCBF-QP controller with YOLO-based sensing and dynamically varying swimmer motion. The red line  $b(x) = 0$  indicates the safe set boundary. Maintaining  $b(x) \geq 0$  corresponds to satisfying the visibility safety constraint throughout the run.

**Tracking Performance.** Beyond single-run visualization, robust filtering is assessed against two baselines: nominal control without safety filtering and non-robust HOCBF filtering. Each variant is run on an identical swimmer trajectory and YOLO sensing configuration. Table I summarizes the quantitative comparison.

TABLE I

QUANTITATIVE COMPARISON ACROSS CONTROLLER VARIANTS. ALL RUNS USE THE SAME SWIMMER TRAJECTORY AND SIMULATION DURATION. YOLO-BASED SENSING IS USED UNLESS NOTED.

| Metric                           | Nominal Only | Non-Robust HOCBF | Robust HOCBF |
|----------------------------------|--------------|------------------|--------------|
| Safe frames (%)                  | 4.081        | 87.129           | 100.000      |
| Violation episodes               | 2            | 3                | 0            |
| Min visibility margin            | -0.205       | -0.205           | 0.000        |
| Mean $ x_{err} $                 | 0.4103       | 0.0667           | 0.0876       |
| RMS $x_{err}$                    | 0.4201       | 0.104            | 0.0951       |
| Mean $ u^* $ (m/s <sup>2</sup> ) | 0.224        | 2.372            | 13.851       |
| Peak $ u^* $ (m/s <sup>2</sup> ) | 15.332       | 20.000           | 20.000       |
| CBF active (%)                   | 0.000        | 9.155            | 6.250        |

Across the same swimmer trajectories and sensing conditions, the robust HOCBF controller yields stronger containment with fewer boundary approaches while maintaining bounded control effort and feasible real-time control. This behavior is consistent with the design objective of enforcing visibility constraints under bounded perception uncertainty.

**Runtime Feasibility.** Real-time feasibility is evaluated from the exported controller runtime logs in the same robust-HOCBF experiment used for safety and tracking plots. Using an 8.333 ms runtime deadline (aligned with 120 Hz frame pacing) and a controller discretization of 0.01 s (100 Hz), the measured control-cycle runtime is 0.0755 ms mean, 0.141 ms at the 95th percentile, and 3.62 ms maximum, with 0.0% over-budget frames. The QP-window portion is

0.00336 ms mean, 0.00420 ms at the 95th percentile, and 0.286 ms maximum, also with 0.0% over-budget frames. These measurements confirm that the robust HOCBF-QP safety filter is computationally lightweight relative to the control period.

## VI. CONCLUSION

This paper presents a perception-aware safe control framework for autonomous swimmer tracking with a rail-mounted camera. By combining YOLO-based detection with robust high-order control barrier functions, the method enforces image-space visibility constraints while respecting rail and actuator limits. The formulation is well matched to this application because the camera dynamics are second order and visibility constraints depend on the projected swimmer bounding box. Offline perception-error analysis provides uncertainty bounds that are incorporated directly into the online safety filter.

Future work includes real-world deployment and improved calibration between image-space detections and rail coordinates, followed by extensions to multi-swimmer tracking and downstream performance-metric extraction.

## REFERENCES

- [1] N. Giuliotti, A. Caputo, P. Chiariotti, and P. Castellini, “SwimmerNET: Underwater 2D swimmer pose estimation exploiting fully convolutional neural networks,” *Sensors*, vol. 23, no. 4, p. 2364, 2023, doi: 10.3390/s23042364.
- [2] Y.-R. Chern *et al.*, “A butterfly stroke swimming recording and performance analysis system based on computer vision and machine learning,” *Measurement*, vol. 251, p. 117171, 2025, doi: 10.1016/j.measurement.2025.117171.
- [3] G. Thomas, R. Gade, T. B. Moeslund, P. Carr, and A. Hilton, “Computer vision for sports: Current applications and research topics,” *Computer Vision and Image Understanding*, vol. 159, pp. 3–18, 2017, doi: 10.1016/j.cviu.2017.04.011.
- [4] E. Ceseracciu, Z. Sawacha, S. Fantozzi, M. Cortesi, G. Gatta, S. Corazza, and C. Cobelli, “Markerless analysis of front crawl swimming,” *Journal of Biomechanics*, vol. 44, no. 12, pp. 2236–2242, 2011, doi: 10.1016/j.jbiomech.2011.06.003.
- [5] F. Ferryanto, A. I. Mahyuddin, and M. Nakashima, “Markerless optical motion capture system for asymmetrical swimming stroke,” *Journal of Engineering and Technological Sciences*, vol. 54, no. 5, p. 220503, 2022, doi: 10.5614/j.eng.technol.sci.2022.54.5.3.
- [6] C. Li, C. Guo, W. Ren, R. Cong, J. Hou, S. Kwong, and D. Tao, “An underwater image enhancement benchmark dataset and beyond,” *IEEE Transactions on Image Processing*, vol. 29, pp. 4376–4389, 2020, doi: 10.1109/TIP.2019.2955241.
- [7] K. Panetta, L. Kezebou, V. Oludare, and S. Agaian, “Comprehensive underwater object tracking benchmark dataset and underwater image enhancement with GAN,” *IEEE Journal of Oceanic Engineering*, vol. 47, no. 1, pp. 59–75, 2022, doi: 10.1109/JOE.2021.3086907.
- [8] J. Chen and M. J. Er, “Dynamic YOLO for small underwater object detection,” *Artificial Intelligence Review*, vol. 57, art. 165, 2024, doi: 10.1007/s10462-024-10788-1.
- [9] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 779–788.
- [10] J. Redmon and A. Farhadi, “YOLO9000: Better, faster, stronger,” in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 7263–7271.
- [11] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, “YOLOv4: Optimal speed and accuracy of object detection,” *arXiv preprint arXiv:2004.10934*, 2020, doi: 10.48550/arXiv.2004.10934.
- [12] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, “YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors,” in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 7464–7475, doi: 10.1109/CVPR52729.2023.00721.

- [13] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of Basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960, doi: 10.1115/1.3662552.
- [14] T. R. Benedict and G. W. Bordner, "Synthesis of an optimal set of radar track-while-scan smoothing equations," *IRE Transactions on Automatic Control*, vol. 7, pp. 27–32, 1962, doi: 10.1109/TAC.1962.1105477.
- [15] F. Chaumette and S. Hutchinson, "Visual servo control. I. Basic approaches," *IEEE Robotics & Automation Magazine*, vol. 13, no. 4, pp. 82–90, 2006, doi: 10.1109/MRA.2006.250573.
- [16] F. Chaumette and S. Hutchinson, "Visual servo control. II. Advanced approaches," *IEEE Robotics & Automation Magazine*, vol. 14, no. 1, pp. 109–118, 2007, doi: 10.1109/MRA.2007.339609.
- [17] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs for safety critical systems," *IEEE Transactions on Automatic Control*, vol. 62, no. 8, pp. 3861–3876, 2017, doi: 10.1109/TAC.2016.2638961.
- [18] X. Xu, P. Tabuada, J. W. Grizzle, and A. D. Ames, "Robustness of control barrier functions for safety critical control," *IFAC-PapersOnLine*, vol. 48, no. 27, pp. 54–61, 2015, doi: 10.1016/j.ifacol.2015.11.152.
- [19] Q. Nguyen and K. Sreenath, "Exponential control barrier functions for enforcing high relative-degree safety-critical constraints," in *Proc. American Control Conference (ACC)*, 2016, pp. 322–328, doi: 10.1109/ACC.2016.7524935.
- [20] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, "Control barrier functions: Theory and applications," in *Proc. European Control Conference (ECC)*, 2019, pp. 3420–3431, doi: 10.23919/ECC.2019.8796030.
- [21] W. Xiao and C. Belta, "Control barrier functions for systems with high relative degree," in *Proc. IEEE 58th Conf. Decision and Control (CDC)*, 2019, pp. 474–479, doi: 10.1109/CDC40024.2019.9029455.
- [22] W. Xiao and C. Belta, "High-order control barrier functions," *IEEE Transactions on Automatic Control*, vol. 67, no. 7, pp. 3655–3662, 2022, doi: 10.1109/TAC.2021.3105491.
- [23] S. Kolathaya and A. D. Ames, "Input-to-state safety with control barrier functions," *IEEE Control Systems Letters*, vol. 3, no. 1, pp. 108–113, 2019, doi: 10.1109/LCSYS.2018.2853698.
- [24] M. Jankovic, "Robust control barrier functions for constrained stabilization of nonlinear systems," *Automatica*, vol. 96, pp. 359–367, 2018, doi: 10.1016/j.automatica.2018.07.004.
- [25] A. Agrawal and K. Sreenath, "Discrete control barrier functions for safety-critical control of discrete systems with application to bipedal robot navigation," in *Proc. Robotics: Science and Systems (RSS)*, 2017, doi: 10.15607/RSS.2017.XIII.073.