Humanoid State Estimation in RoboCup

Thomas O'Brien
University of Newcastle
thomas.obrien@uon.edu.au

***Abstract*—This work presents a comprehensive pipeline for kinematic state estimation of humanoid robots in the RoboCup competition. The dynamic and sensor-limited environment of RoboCup poses significant challenges for accurate state estimation, including unstable walking surfaces, frequent collisions, and restrictions on external sensing modalities like LiDAR and GPS. To address these challenges, we propose an integrated approach that combines odometry, visual localization, and optimization techniques, all designed for real-time performance on resource-constrained hardware.**

I. Introduction

For a humanoid robot, locomotion involves controlling the unactuated floating base to a desired location in the world. Before a control action can be applied, an accurate estimate of the position and orientation of the floating base is required. In the context of Robocup, the artificial grass surface and collisions with other robots complicates stable walking, making state estimation challenging due to falls, noise and drift over time. Furthermore, external sensors that could enhance state estimation, such as LiDAR or GPS, are prohibited in the RoboCup Humanoid League [1]. As a result, robots must rely solely on sensors such as cameras, IMU's and force sensors, adding to the complexity of state estimation. An overview of the commonly used frames of reference used in the Robocup competition is provided in Figure 1. In the context of the Robocup competition, kinematic state estimation can be broken down into two major areas

- **Odometry:** Estimation of the robots pose with respect to an inertial world frame $\mathcal{W}$
- **Localization:** Estimation of the robots pose with respect to the soccer field frame $\mathcal{F}$

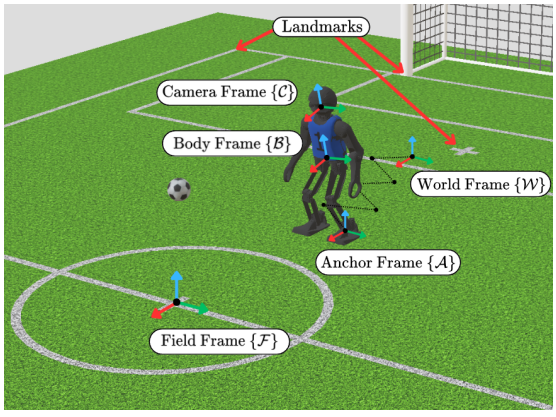

Fig. 1. Common landmarks and frames of reference for state estimation in the Robocup competition

Odometry is often tackled through the use nonlinear observers that fuse leg odometry and Inertial Measurement Unit (IMU) measurements [2]. In this work we present a very simple but practical and effective method for Odometry. Various localization techniques have been developed for RoboCup soccer, including Kalman Filters [3] and Monte Carlo Localization [4]–[6]. These localization methods typically rely on visual landmarks as measurement inputs [7], [8]. While effective, these computer vision pipelines can be computationally expensive, affecting real-time performance. To mitigate this, we present methods for accurate visual landmark detection, which can run in real-time on resource constrained hardware. Additionally, we present a novel approach for localization using the detected visual landmarks, a combination of nonlinear optimization and Kalman filtering.

II. Odometry

Since the contact configuration of a robot during walking is always changing, we construct a representation of the system in a general World-fixed inertial frame. We consider two reference frames: World-fixed ($\mathcal{W}$-frame) inertial frame attached to the ground and body-fixed ($\mathcal{B}$-frame) frame rigidly attached to the robot midway between the robots hip yaw joints. The homogeneous transformation matrix capturing the relationship between these frames is given by

$$\mathbf{H}_b^w = \begin{bmatrix} \mathbf{R}_b^w & \mathbf{r}_{B/W}^w \\ \mathbf{0} & 1 \end{bmatrix} \tag{1}$$

where $\mathbf{R}_b^w \in SO(3)$ is the rotation matrix from the body frame $\mathcal{B}$ to the world frame $\mathcal{W}$, and $\mathbf{r}_{B/W}^w \in \mathbb{R}^3$ is the position vector of the body frame $\mathcal{B}$ with respect to the world frame $\mathcal{W}$. To estimate the orientation of the floating base body frame $\mathcal{B}$, we use the Mahony Filter [9], a simple and efficient approach for real-time attitude estimation. For convenience, the IMU is located near the body frame B. The Mahony filter has only two tuning parameters, the PI compensator gains, K_p and K_i, making the tuning process straightforward. In addition, we estimate the floating base translation $\mathbf{r}_{B/W}^W$ using the anchor point strategy. We select an anchor point A, located on the robots foot sole and assume that this point is grounded at position $\mathbf{r}_{A/W}^W$ in the world frame whenever it serves as the support foot. In the floating-base frame B, the position $\mathbf{r}_{A/B}^B$ of this anchor point is known through forward kinematics allowing continuous tracking of the floating base translation relative to the world frame. Each time the support foot changes during walking, detected using kinematic thresholds on the

relative z height of the feet, the anchor frame is updated to the new support foot. Additionally, this method provides an estimate of the floating-base yaw orientation which is fused with the Mahony filter yaw estimation to reduce drift.

III. Visual landmark detection

Our localization approach relies on visual landmarks detected using two computer vision methods: YOLOv8n [10] and the Visual Mesh [11]. We use YOLOv8n, a state-of-the-art real-time object detection model, to identify objects and key landmarks. Simultaneously, the Visual Mesh serves as a highly efficient semantic segmentation network specifically tuned for detecting field lines. Table I summarizes the features detected by each method. The landmarks used in the localization pipeline include YOLOv8n-detected goal posts, T, L, and X intersections, and field line points detected by the Visual Mesh. To evaluate the computational performance of these approaches, we benchmarked them on two hardware platforms: a simulation laptop equipped with an Intel i7-11850H processor and integrated UHD GPU, and real robot hardware featuring an Intel i7-1260P processor and Iris™ Xe GPU. Table II presents the results, highlighting the achieved frame rates on each platform.

Without loss of generality, through a combination of our camera model and the extrinsic matrix $\mathbf{H}_b^c$, the pixel-based detections can be projected onto the field plane. A detection in world space $\hat{\mathbf{r}}_{O/W}^w \in \mathbb{R}^3$ is given by

$$\hat{\mathbf{r}}_{O/W}^w = \left| \frac{e_3^\top \mathbf{r}_{C/W}^w}{e_3^\top (\mathbf{R}_c^w \mathbf{u}_{O/C}^c)} \right| \mathbf{R}_c^w \mathbf{u}_{O/C}^c + \mathbf{r}_{C/W}^w \tag{2}$$

where $\mathbf{u}_{O/C}^c \in \mathbb{R}^3$ is the unit vector associated with a pixel obtained through our camera model, $\mathbf{r}_{C/W}^w \in \mathbb{R}^3$ is the position of the camera in the world frame, $\mathbf{R}_c^w \in SO(3)$ is the rotation matrix from the camera frame to the world frame, and $e_3 \in \mathbb{R}^3$ is the basis vector $[0, 0, 1]^\top$.

IV. Localization

The localization problem can be formulated as estimating the pose of the field relative to the world frame. Due to the flat nature of the soccer field, this can be fully described by the transformation

$$\mathbf{H}_w^f(\mathbf{x}) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & x \\ \sin\theta & \cos\theta & 0 & y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3}$$

where $\mathbf{x} = [x, y, \theta]^\top \in \mathbb{R}^3$ is a vector containing the x-y translation and yaw rotation. We propose a localization method leveraging nonlinear optimization to compute the optimal state $\mathbf{x}$ in real-time. Nonlinear optimization techniques have recently gained popularity in robot localization due to their superior performance compared to traditional filter-based approaches [12]. Our framework employs the derivative-free algorithm COBYLA [13] (Constrained Optimization BY Linear Approximations), integrating multiple cost components and constraints. The optimization problem is given by

$$\begin{aligned} \mathbf{x}^* = \underset{\mathbf{x}}{\operatorname{argmin}} \quad & J(\mathbf{x}) \\ \text{s.t.} \quad & \mathbf{x}_{\min} \leq \mathbf{x} \leq \mathbf{x}_{\max} \end{aligned} \tag{4}$$

where $\mathbf{x}_{\min}, \mathbf{x}_{\max} \in \mathbb{R}^3$ are the lower and upper bounds on the state vector $\mathbf{x}$. The overall cost function $J(\mathbf{x})$ is defined as

$$J(\mathbf{x}) = w_{\text{fl}} J_{\text{fl}}(\mathbf{x}) + w_{\text{lm}} J_{\text{lm}}(\mathbf{x}) + w_{\text{sc}} J_{\text{sc}}(\mathbf{x}) \tag{5}$$

where $w_{\text{fl}}, w_{\text{lm}}$ and w_{sc} are scalar weights assigned to each component.

Field Line Alignment Cost: $J_{\text{fl}}(\mathbf{x})$ measures how well the observed field line points align with actual field lines, given by

$$J_{\text{fl}}(\mathbf{x}) = \sum_{i=1}^{N_{\text{fl}}} d_{\text{map}} \left(\mathbf{H}_w^f(\mathbf{x}) \hat{\mathbf{r}}_i^w \right)^2 \tag{6}$$

where N_{fl} is the number of observed field line points, $\hat{\mathbf{r}}_i^w \in \mathbb{R}^3$ represents the i-th field line point in the world frame, transformed into the field frame via $\mathbf{H}_w^f(\mathbf{x})^{-1}$, and d_{map} is a function which provides the distance to the nearest field line using a precomputed distance map.

Landmark Cost: $J_{\text{lm}}(\mathbf{x})$ assesses the alignment of observed field line intersections (T, L, and X intersections) and goal posts with known positions on the field, given by

$$J_{\text{lm}}(\mathbf{x}) = \sum_{i=1}^{N_{\text{lm}}} \left\| \mathbf{r}_i^{\text{f}} - \mathbf{H}_w^f(\mathbf{x}) \hat{\mathbf{r}}_i^w \right\|^2 \tag{7}$$

where N_{lm} is the number of associated landmarks, $\mathbf{r}_i^{\text{f}} \in \mathbb{R}^3$ is the known position of the i-th landmark in the field frame, and $\hat{\mathbf{r}}_i^w \in \mathbb{R}^3$ is the observed position of the i-th landmark in the world frame.

State Change Cost: $J_{\text{sc}}(\mathbf{x})$ penalizes significant deviations from the prior state estimate, given by

$$J_{\text{sc}}(\mathbf{x}) = \left\| \mathbf{x} - \mathbf{x}_0 \right\|^2 \tag{8}$$

where $\mathbf{x}_0 \in \mathbb{R}^3$ is the prior state estimate (initial guess), and $\mathbf{x}$ is the current state estimate being optimized. This cost term ensures that the optimizer does not produce abrupt changes in the estimated state between consecutive frames unless strongly supported by the observations.

After each optimization step, the solution $\mathbf{x}$ is filtered using a standard Kalman filter. This filtering step smooths the state estimates over time, improving robustness against noisy observations and enhancing the stability of the localization results. We evaluated the performance of our algorithm using a ground truth dataset collected in the Webots simulation environment [14]. Table III shows the Root Mean Square Error (RMSE) between the estimated poses and the ground truth for various methods and algorithm variations. Notably, our method that incorporates all cost terms with Kalman Filtering achieves the lowest errors. On average, the optimization routine and filtering step take only 2 milliseconds to complete.

REFERENCES

[1] "Robocup humanoid league rules and setup," 2024.

[2] Ross Hartley, Maani Ghaffari Jadidi, Jessy Grizzle, and Ryan M Eustice, "Contact-aided invariant extended kalman filtering for legged robot state estimation," in *Proceedings of Robotics: Science and Systems*, Pittsburgh, Pennsylvania, June 2018.

[3] Michael J. Quinlan and Richard H. Middleton, "Multiple model kalman filters: A localization technique for robocup soccer," in *RoboCup 2009: Robot Soccer World Cup XIII*, Jacky Baltes, Michail G. Lagoudakis, Tadashi Naruse, and Saeed Shiry Ghidary, Eds., Berlin, Heidelberg, 2010, pp. 276–287, Springer Berlin Heidelberg.

[4] Judith Hartfill, *Feature-Based Monte Carlo Localization in the RoboCup Humanoid Soccer League*, Ph.D. thesis, 09 2019.

[5] Stefan Enderle, Marcus Ritter, Dieter Fox, Stefan Sablatnög, Gerhard Kraetzschmar, and Günther Palm, "Vision-based localization in robocup environments," in *RoboCup 2000: Robot Soccer World Cup IV*, Peter Stone, Tucker Balch, and Gerhard Kraetzschmar, Eds., Berlin, Heidelberg, 2001, pp. 291–296, Springer Berlin Heidelberg.

[6] Patrick Heinemann, Juergen Haase, and Andreas Zell, "A combined monte-carlo localization and tracking algorithm for robocup," in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2006, pp. 1535–1540.

[7] Hannes Schulz and Sven Behnke, "Utilizing the structure of field lines for efficient soccer robot localization," *Advanced Robotics*, vol. 26, no. 14, pp. 1603–1621, 2012.

[8] Aislan C. Almeida, Anna H. R. Costa, and Reinaldo A. C. Bianchi, "Vision-based monte-carlo localization for humanoid soccer robots," in *2017 Latin American Robotics Symposium (LARS) and 2017 Brazilian Symposium on Robotics (SBR)*, 2017, pp. 1–6.

[9] R. Mahony, T. Hamel, and J.-M. Pflimlin, "Complementary filter design on the special orthogonal group so(3)," in *Proceedings of the 44th IEEE Conference on Decision and Control*, 2005, pp. 1477–1484.

[10] Rejin Varghese and Sambath M., "Yolov8: A novel object detection algorithm with enhanced performance and robustness," in *2024 International Conference on Advances in Data Engineering and Intelligent Computing Systems (ADICS)*, 2024, pp. 1–6.

[11] Trent Houliston and Stephan K. Chalup, "Visual mesh: Real-time object detection using constant sample density," in *RoboCup 2018: Robot World Cup XXII*, Dirk Holz, Katie Genter, Maarouf Saad, and Oskar von Stryk, Eds., Cham, 2019, pp. 45–56, Springer International Publishing.

[12] Shoudong Huang, "A review of optimisation strategies used in simultaneous localisation and mapping," *J. Control. Decis.*, vol. 6, pp. 61–74, 2018.

[13] M. J. D. Powell, *A Direct Search Optimization Method That Models the Objective and Constraint Functions by Linear Interpolation*, pp. 51–67, Springer Netherlands, Dordrecht, 1994.

[14] Webots, "http://www.cyberbotics.com," Open-source Mobile Robot Simulation Software.

TABLE I
FEATURES DETECTED BY YOLOV8N AND VISUAL MESH

YOLOv8n Features	Visual Mesh Classes
Ball	Ball
Robots	Robots
Goal Posts	Goal Posts
T Intersections	Field Line
L Intersections	Field
X Intersections	Environment

TABLE II
FPS PERFORMANCE OF YOLOV8N AND VISUAL MESH

Method	Sim (i7-11850H) [FPS]	Robot (i7-1260P) [FPS]
YOLOv8n	47	66
Visual Mesh	152	259

TABLE III
RMSE ERROR BETWEEN ESTIMATED AND GROUND TRUTH POSE IN ROBOCUP WEBOTS SIMULATION ENVIRONMENT

Method	x [m]	y [m]	yaw [deg]
Particle Filter	0.0563	0.0890	1.6180
NLopt (field lines only)	0.0503	0.0563	0.8389
NLopt (field intersections only)	0.0629	0.0617	1.7125
NLopt (goal posts only)	0.9085	0.1239	2.5690
NLopt (all cost terms without KF)	0.069	0.0995	0.8730
NLopt (all cost terms)	**0.0500**	**0.0559**	**0.8273**