Cooperative Inchworm Localization with a Low Cost Heterogeneous Team

by Brian Nemsick

Research Project

Submitted to the Department of Electrical Engineering and Computer Sciences, University of California at Berkeley, in partial satisfaction of the requirements for the degree of **Master of Science, Plan II**.

Approval for the Report and Comprehensive Examination:

Committee:

Professor A. Zakhor
Research Advisor

(Date)

* * * * * * *

Professor R. Fearing
Second Reader

(Date)

Cooperative Inchworm Localization with a Low Cost Heterogeneous Team

Brian Nemsick
Advisor: Avideh Zakhor
Department of Electrical Engineering and Computer Science
University of California, Berkeley

Abstract

In this thesis we address the problem of multi-robot localization with a heterogeneous team of low-cost mobile robots. The team consists of a single centralized "observer" with an inertial measurement unit (IMU) and monocular camera, and multiple "picket" robots with only IMUs and Red Green Blue (RGB) light emitting diodes (LED). This team cooperatively navigates a visually featureless environment such as a collapsed building. A combination of camera imagery captured by the observer, as well as IMU measurements on the pickets and observer are fused to estimate motion of the team. A team movement strategy, referred to as inchworm, is formulated as follows; pickets move ahead of the observer and then act as temporary landmarks for the observer to follow. This cooperative approach employs a single Extended Kalman Filter (EKF) to localize the entire heterogeneous multi-robot team, using a formulation of the measurement Jacobian to relate the pose of the observer to the poses of the pickets with respect to the global reference frame. An initial experiment with the inchworm strategy has shown localization within 0.14 m position error and 2.18° orientation error over a path-length of 5 meters in an environment with irregular ground, partial occlusions and a ramp. This demonstrates improvement over a camera only relative pose measurement localization technique that was adapted to our team dynamic which produced 0.18m position error and 3.12 ° orientation error over the same dataset.

1 Introduction

The size of a robot can greatly affect what it can do and where it can go. Advantages of small robots include increased accessibility and a wider range of capabilities such as crawling through pipes, inspecting collapsed buildings, exploring congested or complex environments, and hiding in small or inconspicuous spaces. However, these benefits also bring along challenges in the form of limited energy availability, reduced sensing abilities, lower communication capability, limited computational resources, and tighter power constraints.

One way to overcome these limitations is to employ a heterogeneous team [1] of collaborative robots. This approach marks a design shift away from the traditional simultaneous localization and mapping (SLAM) ground robots that have expensive sensors and powerful processors, but less mobility in disaster environments. The goal is to have small, mobile, disposable robots with limited capabilities collaborate and share information to accomplish a larger task. Since each robot is expendable, reliability can be obtained in numbers because even if a single robot fails, few capabilities are lost for the team. Hierarchical organization and the idea of a heterogeneous teams allows for robots with different specializations, such as larger robots with higher computation power, smaller robots with increased maneuverability, robots with different sensor modalities, and so on. Another advantage of a team of less capable robots, rather than one extremely capable robot, is that it allows sensing from multiple viewpoints and hence a wider effective baseline. This is helpful for tasks such as surveillance, exploration, and monitoring. Furthermore, physically traversing an area conveys much more information than simply looking at it from a distance. For example, an expensive scanner can scan the rubble of a disaster site from the outside, but cannot enter and inspect the inside. Knowledge that cannot be gained without physical presence includes detection of slippery surfaces, hidden holes, and other obscured hazards; these can completely incapacitate robots despite their state-of-the-art SLAM algorithms, expensive cameras, and complex laser range finders. Instead, these same hazards can be detected through sacrifice of highly mobile disposable picket robots that scout the area [2].

Localization is a central problem in many robotic applications. It is particularly important in collaborative situations where, without position and orientation information of each robot, there is no global context in which to meaningfully share the information between the robots. In this thesis, we focus on the localization problem and consider a heterogeneous multi-robot team consisting of two types of minimally equipped robots. A single, central, and more capable observer robot is equipped with a monocular camera and a 6-axis IMU consisting of a gyroscope and accelerometer. Multiple picket robots, which are expendable and less computationally capable, are equipped with no sensors other than 6-axis IMUs. A limited communication interface between the observer robot and individual picket robots is assumed. We consider a multi-robot team with a single observer and multiple picket robots in an unknown environment. We present a method for using a single Extended Kalman Filter (EKF), which the observer uses to localize the entire multi-robot team, including itself, in six degrees of freedom (6-DOF) by fusing IMU measurements and relative pose estimates of the pickets. Relative pose estimation refers to the process of estimating the position and orientation of a picket's body frame with respect to the camera frame on the observer. Red Green Blue (RGB) light emitting diodes (LED) are mounted at known positions on the picket robots to enable relative pose estimation.

In this thesis we present a solution to 6-DOF localization in unstructured, non-planar, and low light environments without GPS, preplaced landmarks, external visual features, and reliable wheel odometry for a team consisting of a single observer robot and multiple picket robots. The team processes data asynchronously, meaning that IMU measurements and pose estimates are not received in temporal order, because of communication latency. This approach and team dynamic is targeted for environments such as collapsed building where large robots with greater sensor capabilities may not be able to traverse the environment. In Section 2, we describe the approach of cooperative inchworm localization. The cooperative inchworm localization approach is a turn based strategy where at any giving point of time at least one robot remains stationary while the other

robot move. Specifically, we combine the turn based strategy movement and IMU motion models to reduce IMU dead-reckoning error. We add multiple color LED to facilitate relative pose estimation of the pickets with respect to the observer. We derive the measurement model specific to the observer-picket team dynamic which is used to compute the EKF update.

The datasets discussed in Section 3, results of this thesis include one observer working together with two pickets to traverse given areas. Even with minimal sensors, the inchworm method is shown to work in dark environments with visual occlusions such as walls or obstacles, instances when line of sight between the robots is lost, and nonplanar settings without external visual features or landmarks. The camera and IMU fusion approach employed by the inchworm method demonstrates improved performance over a camera only approach. In addition, the accuracy of the approach improves with number of picket robots. Conclusions and future work are included in Section 4.

1.1 Related Work

Existing strategies using stationary robots have been employed [1]. A stationary robot is defined as a robot that remains at rest while the other robots move. Stationary robots and leapfrogging strategies build on the ideas from [1] and have shown promise in 3-DOF environments in [3][4]. These previous approaches have a stronger condition of requiring two or three stationary robots at any given time. The inchworm strategy relaxes these constraints to require only a single stationary robot as shown in Figure 1. As shown in Figure 1 (a-c), the leap frog method uses a homogeneous team where each robot is the same. At each time step one robot moves while the other two remain stationary. For example during (a) at t = 1, robots 2 and 3 remain stationary while robot 1 moves. This process repeats where the moving robot cycles at each time step. The inchworm strategy requires at least one robot to remain stationary. In addition the picket robots generally remain in front of the observer. For example during Figure 1 (d) the pickets move in front of the observer and during (e) the observer catches up to the stationary pickets. At (f) picket-1 and the observer both move while picket-2 is stationary.

A similar approach, cooperative positioning system (CPS), to inchworm is presented in [5]. The CPS approach focuses on 4-DOF (x,y,z,yaw) environments and partitions the robots into two groups each consisting of at least one robot. Under the CPS system, the team alternates between which group moves. Either group A moves or group B moves. For the purposes of comparison we can consider group A to be the observer and group B to be the picket robots. An example of the CPS motion is shown in Figure 1 (d-e) where either the pickets more or the observer moves. Our inchworm strategy improves on CPS by allowing the observer and picket robots to move at the same time. An example of this is found in Figure 1 (f) where both picket-1 and the observer move while picket-2 is stationary. This adds a higher degree of flexibility in team movement. Previous approaches and variants of leapfrogging strategies were focused on team dynamics with high redundancy where each robot produces relative pose estimates. The inchworm approach relaxes the sensor constraints of the team to accommodate teams where a single observer robot is required to have relative pose estimation capabilities. This leaves the picket robots with more flexibility and specialization.

Haldane et al. [2] use a heterogeneous team to detect slippery terrain by sending out a small picket robot and having it walk around the area of interest. The large robot is capable of accurately estimating it's own pose. The large robot uses an augmented reality (AR) tag on the picket robot for localization. Then, features of the picket's motion are used to train a terrain classifier capable of detecting slippery terrain. This approach demonstrates the value of picket robots in a heterogeneous robot team for exploration or SLAM.

A follow-the-leader approach in [6] demonstrates a team composition similar to picket-observer. The leaders and children setup in [7] provides a relative localization scheme in 3-DOF; it assumes accurate localization of the leaders from an external source and localizes of the children robots. This approach is extended in [8] to localize the leaders. The problem is subdivided into leader localization and then children localization. The

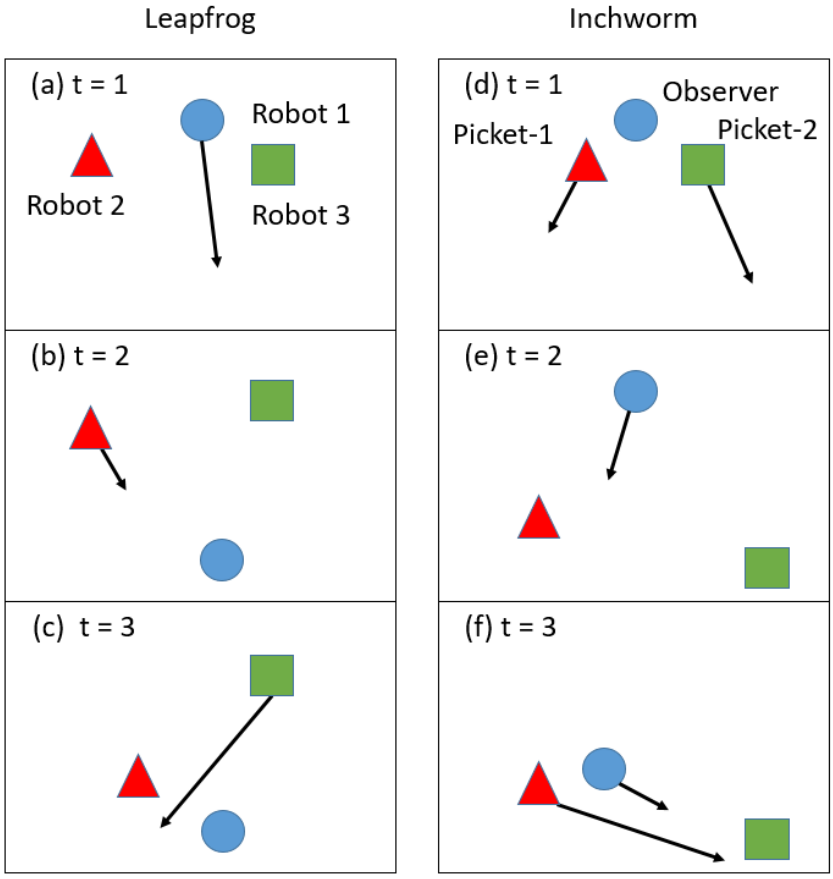

Figure 1: The above diagrams compare the leapfrog and inchworm strategies. Arrows are drawn to show motion that happens during a time step. In the Leapfrog method (a-c), all robots are the same type and at each time step one robot moves while the other two remain stationary. For example during (a) at t = 1, robots 2 and 3 remain stationary while robot 1 moves. This process repeats where the moving robot cycles at each time step. In our approach, the inchworm method, at least one robot remains stationary while two move. In addition the picket robots generally remain in front of the observer. For example during (d) the pickets move in front of the observer and during (e) the observer catches up to the stationary pickets. At (f) picket-1 and the observer both move leaving picket-2 stationary.

localization of the leaders in [8] requires multiple leader to maintain line of sight to each other. Line of sight refers to an unobstructed path between a sensor and a robot that is inside the sensor's field of view. We extend the approach in [8] to jointly solve the leader and children localization problem without requiring multiple leaders.

Alternatives to cooperative localization methods have been demonstrated with Khepera robots [9]. They are 5cm in diameter, modular, and can support various sensors. They rely on dead reckoning, global positioning system (GPS) or fixed global cameras to localize. For unknown, low-light and potentially indoor environments GPS is not reliable and using/mounting a global camera may not be possible. Other small-scale cooperating robots have been demonstrated in [10]. They address the localization and coordination problem with a centralized controller and global camera.

A more recent approach [11] uses range-only sensors with a team of aerial vehicles for SLAM and builds on the limited sensor approach of [12]. These drones are equipped with on-board computers and lasers. Our approach is to use inexpensive and disposable picket robots in 6-DOF environment.

Limited communication capabilities between robots are addressed by [13][14], and this is increasingly critical in multi-robot teams. Ants robots [15] are small and demonstrate localization capabilities with encoders under these constraints. In featureless non-planar 6 DOF enviroments these encoders introduce drift from wheel slippage.

Odometry-based propagation have been successful in 3-DOF fusion architectures [1] [16]. In 6-DOF non-planar environments, wheel slippage causes systematic biases from encoders. Cell phone quality IMUs are a low cost alternative to wheel encoders in 6-DOF environments because they provide a motion model even under slippage. Extensive work in IMU-based propagation in visual-inertial systems has been explored in [17][18][19]. Monocular pose estimation has been explored in [20][21].

Many algorithms and approaches exist for multi-robot localization. A least-squares optimization approach to localization with a multi-tier heterogeneous team was employed in [1], which used infrared (IR), GPS, and sonar modules to produce an estimate at each time step. Graph based approaches have also been used [24][25], and the graph optimization algorithm in [24] relies on the locations of static landmarks and exploits the sparse nature of the graph. The use of cluster matching algorithms with IR sensors is presented in [26]. Maximum a posteriori approaches have also been used for multi-robot localization [27] and are robust to single-point failures. Existing EKF [28][29][16] or particle filter methods [30][13][31] demonstrate the capability of fusing data to provide accurate multi-robot localization.

The noted previous works have extensively explored multi-robot localization, their experiments were conducted with access to significantly more capable robots, availability of GPS or beacons of known pose, 3-DOF settings with planar environmental assumptions and accurate wheel odometry, requirements of additional stationary robots, assumptions of light, and existence of landmarks or visual features. In this thesis we relax these assumptions to localize a team consisting of a single observer robot and multiple picket robots. This is accomplished using an EKF approach with the inchworm strategy requirement of at least a single stationary robot at all times. IMU measurements are used for EKF propagation and relative pose estimates are used as an EKF update.

2 Approach

General Notation	**Example**
Scalar: lower case	x
Vectors: lower case bold	$\mathbf{x}$
Matrices: upper case bold	$\mathbf{X}$
Identity Matrix	$\mathbf{I}$
Zero Matrix	$\mathbf{0}$
Time derivative: dot	$\dot{\mathbf{p}} = \mathbf{v}$
Unit quaternion: over-bar	$\bar{\mathbf{q}}$
Quaternion product: $\otimes$	$\mathbf{q}_1 \otimes \mathbf{q}_2$

Estimation Notation	**Example**
Estimated value: hat	$\hat{\mathbf{x}}$
Estimation error: tilde	$\tilde{\mathbf{x}}$
Rotation correction:	$\delta\boldsymbol{\theta}$
Linear correction: Δ	$\Delta\mathbf{p}$
Rotation error: $\tilde{\boldsymbol{\theta}}$ or $\delta\mathbf{q}$	${}^{G}\tilde{\boldsymbol{\theta}}$ or $\delta\mathbf{q}$
Linear error definition	$\tilde{\mathbf{p}} = \mathbf{p} - \hat{\mathbf{p}}$
Rotation error definition	$\delta\mathbf{q} = \mathbf{q} \otimes \hat{\mathbf{q}}^{-1}$

Coordinate Frame Notation	**Example**
Coordinate frame: upper case	G or {G}
Point in a coordinate frame: point left subscript, frame left superscript	${}^{G}_{B}\mathbf{p}$
Rotation matrix	$\mathbf{R}$
Coordinate frame rotation: from left subscript to left superscript	${}^{B}_{G}\mathbf{R}$ or ${}^{B}_{G}\mathbf{q}$

Subscript Notation	**Example**
Picket robot index, i: right subscript	$\mathbf{x}_i$
Observer robot index, o: right subscript	$\mathbf{x}_o$
General robot index, r: right subscript	$\mathbf{x}_r$
Time index, k: right sub-subscript	${}^{B}_{C}\mathbf{p}_{i_k}$

Coordinate Frames	**Example**
Global:	G or {G}
Observer Body:	O or {O}
Picket Body:	B or {B}
Camera:	C or {C}

EKF Symbols	Definition
$\mathbf{\Sigma}$	covariance matrix
$\mathbf{K}$	Kalman gain
$\mathbf{H}$	observation matrix
$\mathbf{H}_i$	observation matrix with respect to the i^{th} picket robot
$\mathbf{I}$	identity matrix
$\mathbf{n}$	Gaussian noise vector
$\mathbf{Q}$	observation noise matrix
$\mathbf{Q}_i$	observation noise matrix with respect to the i^{th} picket robot
$\mathbf{r}$	residual
$\mathbf{x}$	state vector
$\Delta\mathbf{x}$	state correction
$\Delta\mathbf{x}_r$	state correction with respect to robot r
$\mathbf{z}$	observation
$\mathbf{z}_i$	observation with respect to the i^{th} picket robot

State Symbols	Definition
n	total number of picket robots
${}^{O}_{G}\bar{\mathbf{q}}$	rotation from global to observer body frame
${}^{G}_{B}\bar{\mathbf{q}}_i$	rotation from global to the i^{th} picket body frame
${}^{G}\mathbf{p}_r$	position in the global frame with respect to robot r
${}^{G}\mathbf{v}_r$	velocity in the global frame with respect to robot r
$\mathbf{b}_{r_g}$	bias of the gyroscope with respect to robot r
$\mathbf{b}_{r_a}$	bias of the accelerometer with respect to robot r
${}^{G}\tilde{\boldsymbol{\theta}}_r$	orientation error in the global frame with respect to robot r

IMU Symbols	Definition
$\boldsymbol{\omega}$	rotational velocity vector
$\mathbf{a}$	acceleration vector
$\mathbf{n}_g$	gyroscope white noise
$\mathbf{n}_a$	accelerometer white noise
$\mathbf{n}_{wg}$	gyroscope bias driving white noise
$\mathbf{n}_{wa}$	accelerometer bias driving white noise

Pose Estimation Symbols	Definition
Σ	pixel noise
$\mathbf{C}$	LED correspondence
$\mathbf{d}$	set of LED centroids
D	number of LED centroid detections
$\mathbf{J}$	Jacobian of Gauss-Newton minimization on reprojection error
$\mathbf{l}$	set of LED configuration
L	number of LEDs on a robot
P	pose estimate
$[{}^{C}_{O}\bar{\mathbf{q}}^T, {}^{C}_{O}\mathbf{p}^T]^T$	static camera transform from observer body frame

The purpose of the multi-robot EKF is to localize all of the robot team's body frames with respect to a global reference frame. Each robot is part of the EKF state as described in Section 2.1 An overview of the EKF is provided in Algorithm 1 and can be described as follows: IMU measurements from both types of robots are used to propagate the state and covariance of the team with the same IMU motion model. RGB LEDs are placed with a known configuration on each picket robot such that images captured on the observer can be used to estimate the relative pose of the robots using [22][23] and Gauss-Newton minimization[20]. Relative pose is defined as the estimation of a picket's body frame, position and orientation, with respect to the camera frame on the observer. The coordinate frames of the team and example LED placement scheme are depicted in Figure 2. The relative pose estimates are subsequently used in the EKF update step.

The block diagram for the asynchronous data processing is depicted in Figure 3. Each robot in the mobile systems block receives velocity commands from a controller that dictate the motion of the robot. In practice this could be a user manually driving the robots or feedback control on the state estimates. The host system computes relative pose estimate and the EKF. Currently the host is an external laptop but could be the observer robot in the future. Each robot sends IMU measurements and whether it is moving to the host system sensor data buffer. In addition, the observer robot transmits camera imagery to the host system to estimate relative pose. On the host system, an array of n pose estimators compute the relative pose for n picket robots. The relative pose estimates are then put into the sensor data buffer. The sensor data buffer ensures that messages can be processed synchronously, in temporal order, despite being received asynchronously. The period at which the sensor data buffer is processed is addressed in detail in Section 2.5. The buffer is processed in temporal order, emptied, on a fixed period and the EKF computation is performed with the synchronous data for each robot according to Algorithm 1. After the data is consumed for EKF computation it is discarded, thus emptying the buffer for the next period.

A team movement strategy called inchworm is adopted, where the picket robots move ahead of the observer to scout and then the observer robot catches up. This movement strategy requires at least one stationary robot. This turn based approach significantly reduces IMU dead-reckoning error and increases the robustness of the localization algorithm to temporary line of sight as shown in Section 3. An inchworm increment is a set of motions where the observer and picket robots all move at least once. An example inchworm increment is shown in Figure 1 (d-e). A stationary robot does not propagate it's corresponding states or covariances, thus bounding the uncertainty of the entire team. This enables the stationary robot to function as a temporary visual landmark and serves as a functional substitute to external visual features. Although external visual features are

Algorithm 1: Cooperative Inchworm Localization (EKF)

Propagation: For each IMU measurement:

- buffer previous IMU measurements received from other robots
- propagate state and covariance for the team using the time-step, buffer and new IMU measurement (cf. Section 2.2).

Update: For each camera image:

- identify RGB LEDs (cf. Section 2.3.1).
- estimate the relative pose between the visible picket robots and the observer frame with P3P and Gauss-Newton minimization (cf. Section 2.3.1).
- propagate the state and covariance for the team using the time-step, and most recent IMU measurements (cf. Section 2.2)
- perform state and covariance update for the team (cf. Section 2.3.2,2.4).

Inchworm requirement: At least one stationary robot at all times

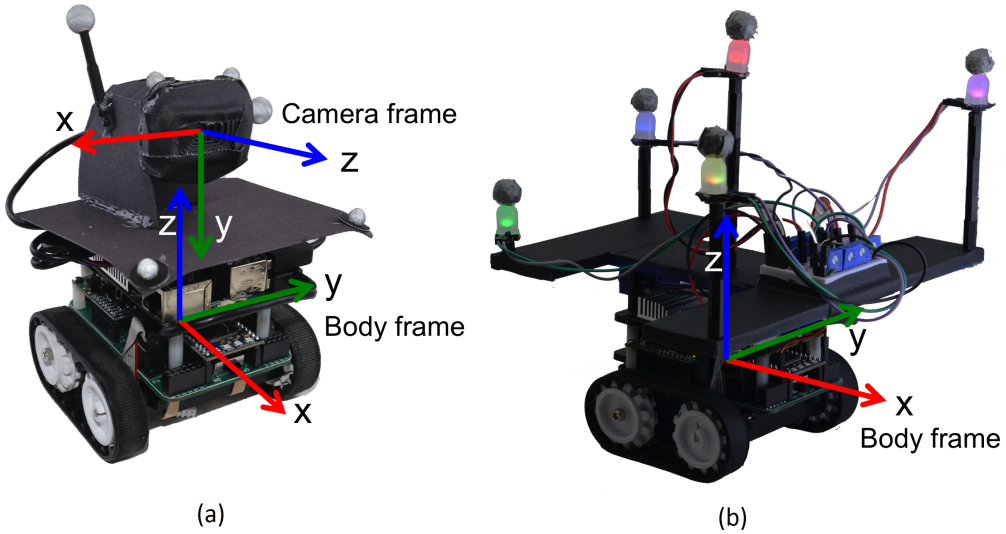

Figure 2: Coordinate frame overview for a sample team consisting of two robots. The observer, (a), is mounted with a camera and the picket, (b), with multi-color LED markers.

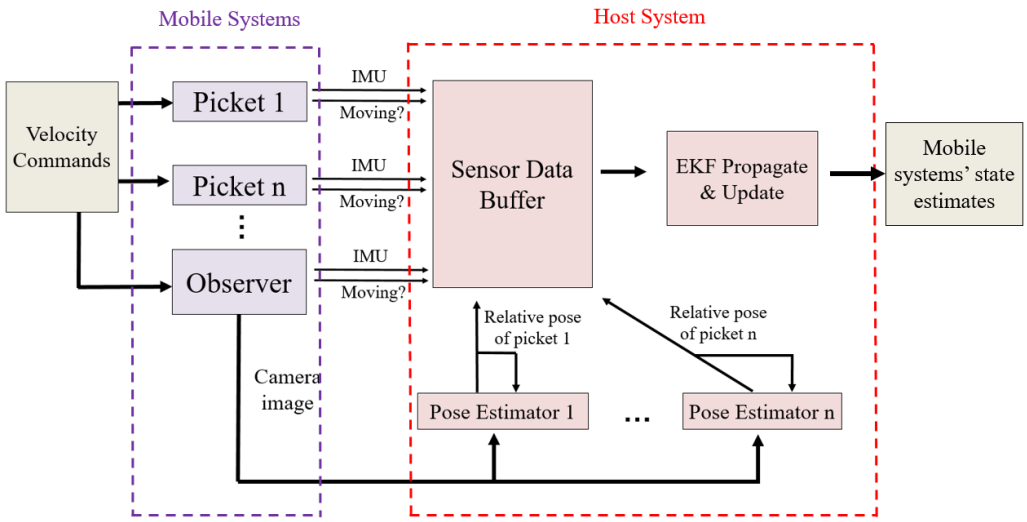

Figure 3: Block diagram of the asynchronous multi-robot team performing real-time cooperative localization algorithm. Asynchronous sensor data from the robots is sent over WiFi, sorted into a measurement buffer, and then used in the EKF propagate and update step. Currently, the host system is an external laptop.

used in traditional visual odometry or visual SLAM systems, they are not consistently available in low light environments.

One benefit of a stationary picket robot is in situations of complete line of sight failure, where none of the picket robots are visible to the observer. In this case, a single future re-observation of a stationary robot, i.e. loop-closure, corrects the IMU dead-reckoning error of the non-stationary robots.

The following sections describe the EKF propagation and update steps in detail. Specifically, Section 2.1 describes the EKF state vector, Section 2.2 describes the IMU motion model, Section 2.3 describes the camera measurement model, Section 2.4 describes the EKF update and Section 2.5 describes the procedure

for handling asynchronous measurements. A quaternion reference is provided in Appendix A and an EKF algorithm reference is provided in Appendix B.

2.1 State Space Representation

The EKF state and accompanying error-state vector stores the state of each single-robot in the multi-robot team. The state vector components with respect to i^{th} picket robot are:

$$\mathbf{x}_i = [{}^B_G\bar{\mathbf{q}}_i^T, {}^G\mathbf{p}_i^T, {}^G\mathbf{v}_i^T, \mathbf{b}_{i_g}^T, \mathbf{b}_{i_a}^T]^T \in \mathbb{R}^{16\times 1} \tag{1}$$

where ${}^B_G\bar{\mathbf{q}}_i^T \in \mathbb{R}^{4\times 1}$, is the unit quaternion representation of the rotation from the global frame {G} to the body frame {B}, ${}^G\mathbf{p}_i, {}^G\mathbf{v}_i \in \mathbb{R}^{3\times 1}$ are the body frame position and velocity with respect to the global frame, and $\mathbf{b}_{i_g}, \mathbf{b}_{i_a} \in \mathbb{R}^{3\times 1}$ are the gyroscope and accelerometer biases.

The corresponding error-state components with respect to i^{th} picket robot are:

$$\tilde{\mathbf{x}}_i = [{}^G\tilde{\boldsymbol{\theta}}_i^T, {}^G\tilde{\mathbf{p}}_i^T, {}^G\tilde{\mathbf{v}}_i^T, \tilde{\mathbf{b}}_{i_g}^T, \tilde{\mathbf{b}}_{i_a}^T]^T \in \mathbb{R}^{15\times 1} \tag{2}$$

where ${}^G\tilde{\boldsymbol{\theta}}_i^T$ is the minimal representation from the error quaternion $\delta\bar{\mathbf{q}} \simeq [\frac{1}{2}{}^G\tilde{\boldsymbol{\theta}}^T, 1]^T$ [18] [19]. The non-quaternion states use the standard additive error model. For example, $\tilde{\mathbf{p}} = \mathbf{p} - \hat{\mathbf{p}}$ where $\hat{\mathbf{p}}$ is the estimated position and $\mathbf{p}$ is the true position. This results in a reduction by 1 of the dimensions $\mathbf{x}_i$ as compared with $\tilde{\mathbf{x}}_i$.

The observer robot is also a component in the EKF state and error-state vector:

$$\begin{aligned} \mathbf{x}_o &= [{}^O_G\bar{\mathbf{q}}_o^T, {}^G\mathbf{p}_o^T, {}^G\mathbf{v}_o^T, \mathbf{b}_{o_g}^T, \mathbf{b}_{o_a}^T]^T \in \mathbb{R}^{16\times 1} \\ \tilde{\mathbf{x}}_o &= [{}^G\tilde{\boldsymbol{\theta}}^T, {}^G\tilde{\mathbf{p}}_o^T, {}^G\tilde{\mathbf{v}}_o^T, \tilde{\mathbf{b}}_{o_g}^T, \tilde{\mathbf{b}}_{o_a}^T]^T \in \mathbb{R}^{15\times 1} \end{aligned} \tag{3}$$

where $\{O\}$ denotes the observer frame.

Combining the states in Eqns. 1, 2, and 3, the augmented EKF state vector and error-state vector with respect to the multi-robot team with n pickets becomes:

$$\mathbf{x} = [\mathbf{x}_o^T, \mathbf{x}_1^T, \mathbf{x}_2^T, ... \mathbf{x}_{\text{n}}^T]^T \in \mathbb{R}^{16(n+1)\times 1}, \tilde{\mathbf{x}} = [\tilde{\mathbf{x}}_o^T, \tilde{\mathbf{x}}_1^T, \tilde{\mathbf{x}}_2^T, ... \tilde{\mathbf{x}}_{\text{n}}^T]^T \in \mathbb{R}^{15(n+1)\times 1} \tag{4}$$

where *n* is the total number of picket robots.

2.2 IMU Propagation Model

The EKF propagation step occurs each time a new IMU measurement from any single-robot or a camera image is captured on the observer robot. Over time as the team performs inchworm increments, the picket and observer robots states become correlated. This correlation is preserved under IMU propagation despite the propagation being calculated independently for each robot. The observer and picket robots utilize the same linearized motion model. An IMU buffer is used to handle asynchronous IMU measurements as described in Section 2.5.

The continuous dynamics of the IMU propagation model for a single-robot are [18][19]:

$$\begin{aligned} {}^B_G\dot{\mathbf{q}} &= \frac{1}{2}\boldsymbol{\Omega}(\boldsymbol{\omega}){}^B_G\bar{\mathbf{q}} \\ {}^G\dot{\mathbf{p}} &= {}^G\mathbf{v} \\ {}^G\dot{\mathbf{v}} &= {}^G\mathbf{a} \\ \dot{\mathbf{b}}_g &= \mathbf{n}_{wg} \\ \dot{\mathbf{b}}_a &= \mathbf{n}_{wa} \end{aligned} \tag{5}$$

where $\mathbf{n}_{wg}, \mathbf{n}_{wa}$ are Gaussian white noise vectors for the gyroscope and accelerometer respectively, ${}^{G}\mathbf{a} = [\mathrm{a}_x, \mathrm{a}_y, \mathrm{a}_z]^T$ is the acceleration in the global frame, $\boldsymbol{\omega} = [\omega_x, \omega_y, \omega_z]^T$ is the rotational velocity in the body frame, and

$$\boldsymbol{\Omega}(\boldsymbol{\omega}) = \begin{bmatrix} -\lfloor \boldsymbol{\omega} \times \rfloor & \boldsymbol{\omega} \\ -\boldsymbol{\omega}^T & 0 \end{bmatrix} \tag{6}$$

$$\lfloor \boldsymbol{\omega} \times \rfloor = \begin{bmatrix} 0 & -\boldsymbol{\omega}_z & \boldsymbol{\omega}_y \\ \boldsymbol{\omega}_z & 0 & -\boldsymbol{\omega}_x \\ -\boldsymbol{\omega}_y & \boldsymbol{\omega}_x & 0 \end{bmatrix} \tag{7}$$

The discrete time linearized model and the error-state model are derived and discussed with detail in [18][19]. Under this model, IMUs have an additional sensor measurement parameter noise for the gyroscope and accelerometer defined as $\mathbf{n}_g$ and $\mathbf{n}_a$ respectively. The parameters $\{\mathbf{n}_g, \mathbf{n}_a\}$ model the measurement noise of the IMU as white noise and $\{\mathbf{n}_{wg}, \mathbf{n}_{wa}\}$ model the biases as Brownian motion processes with derivatives driven by white noise. [18][19]. The IMU values used for our specific robots are discussed in Section 3.1.1.

Critically, stationary robots receive no state or covariance propagation. For example, if the i^{th} picket robot is stationary between time k and $k+1$: $\hat{\mathbf{x}}_{i_{k+1}} = \hat{\mathbf{x}}_{i_k}$. This prevents IMU dead-reckoning drift from moving a temporary landmark and maintains a bounded covariance block pertaining to the stationary robot.

2.3 Camera Measurement Model

In this section we describe the measurement model of the EKF. In Section 2.3.1 we describe the processing used to compute relative pose estimates from camera imagery and discuss the benefits and challenges of using multi-color LEDs on each robot. The relative pose estimates feed directly into the residual and observation matrix derivation used to compute the EKF update in Section 2.3.2.

2.3.1 Relative Pose Estimation

In this section we describe the process of using a camera image to estimate relative pose from observer to visible pickets. In this context, relative pose estimation refers to estimating the pose of a visible picket robot with respect to the camera coordinate frame on board the observer. A relative pose estimate consists of a pose, quaternion and position in the camera frame, with covariance that are directly used to calculate a residual and observation matrix which are used in EKF update.

As shown in Figure 2 (b).Four or more active markers, RGB LEDs, are placed at known configurations on the picket robots to enable relative pose estimation on board the observer. An overview of the relative pose estimation system is in Figure 4.

The first step consists of LED detection. The camera image is converted to the HSV color space where the hue and saturation is thresholded. As described in detail in [20], the LED detection has two distinct modes. In initialization, the entire image is searched for LEDs. Once a single pose estimate has been found from a previous image, LED detection occurs in a bounding box. An example of these bounding boxes is shown in Figure 5. The blue boxes represent the bounding boxes, LED detections are marked with crosshairs of the LED color, and the body frame is overlaid. The bounding box is responsible for feedback loop between "LED detection" and "pose with covariance" blocks in Figure 4. The key point is that given a previous pose estimate, computation time can be saved by performing pose estimation on a smaller portion of an image as defined by the bounding box. The procedure for computing the bounding box and tracking the LEDs in the image frame is addressed in [20].

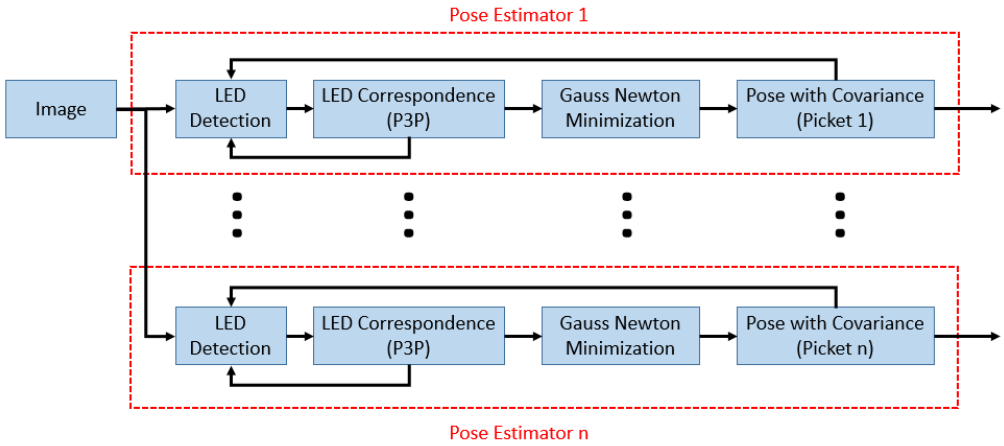

Figure 4: Multi-robot pose estimation for a team consisting of n robots.

A major design decision has to do within the LED correspondence step. The relative pose estimator was adapted to support multi-color LED configuration rather than single-color or infrared [20]. The first key benefit is that it increases accuracy and precision of the pose estimator at further ranges. A persistent issue when using single-color LEDs is that as the picket robot's distance from the camera increases the LED correspondence problem becomes harder and inevitably causes repeated errant measurements. This is a direct consequence of the pixel coordinate locations of the LED detection becoming closer. Multi-color LEDs have occasional problems with LED correspondence because each LED has a different color. Using multiple colors is especially valuable when partial occlusions occur and not all LEDs are visible. Single-color LEDs can run into degenerate geometries under partial occlusions.

In scaling to multiple robots, a multi-color approach eventually has to share colors between robots while this is not an issue with single-color approach. This is potentially problematic for the multi-color approach. For the first camera image, the simplest approach is exhaustive searching the LED to robot correspondence. To avoid exhaustive search in the initialization step, alternative approaches such as power cycling the LEDs in known patterns or motion patterns could be used as viable alternatives. An example motion pattern is to keep one picket stationary, and move all of the others such that the stationary LEDs belong to the appropriate robot. In practice, for small teams, $n < 5$, this is a non-issue for the multi-color approach. Built into the relative pose estimation approach from [20] is bounding boxes and tracking in the image plane to reduce computation in subsequent images. As shown by the blue boxes in Figure 5, the pose estimator only checks for LEDs inside the bounding box. Once the picket robot for a given pose estimator is found for the first time it ignores the LEDs from other robots in subsequent frames. This is shown by the non-overlapping blue bounding boxes in Figure 5 along with the coordinate frames and LED detections.

The second key benefit of the multi-color approach is the computational complexity of the pose estimation from prospective-3-point P3P algorithm [22] [23]. The computational complexity of the LED correspondence step in the single-color approach is given by [20]:

$$4 \cdot \binom{D}{3} \frac{L!}{(L-3)!} \tag{8}$$

where L is the number of LEDs on the robot and D is the number of LED centroid detections. The computational complexity of the LED correspondence step in the multi-color approach is given by:

$$4 \cdot \binom{D}{3} \tag{9}$$

Starting with Eqns. 8 we can directly derive Eq. 9 as follows. The first computational step in in Eq. 8 is to find every combination of 3 LEDs [20]. We call a combination of 3 LEDs, an LED cluster. This result in the $\binom{D}{3}$ term from Eq. 8 and 9. The next computational is checking all permutations of each LED cluster [20]. The LED correspondence of a cluster, which LED is which on the robot, is unknown in the single-color case. Hence, all permutations of 3 LEDs, $\frac{L!}{L-3!}$, must be checked. In the multi-color case, each LED on a robot has a known unique color. We know the configuration directly without enumerating exhaustively through $\frac{L!}{L-3!}$ configurations. The constant factor of 4 in Eqns. 8 and 9 comes from the 4 solutions in the P3P algorithm [22] [23] which is common to both single-color and multi-color approaches. To implement the multi-color approach we modified the source code from [20] as follows. The first change was the multi-color LED thresholding previously described with a hue and saturation threshold. The second change was the direct solution with multi-color LED correspondence. Thus, the correspondence search of our multi-color approach is $\frac{L!}{L-3!}$ times faster than the single color approach.

With large number of robots the first image is computationally expensive. This is because without the first pose as a feedback loop, the initialization state of the LED detection passes all of the LEDs in the image rather than a subset that pertain to the appropriate robot. During this first image, the multi-color naive exhaustive search correspondence step has computational complexity of $D \approx L \cdot n$ from Eq. 9 whereas single-color is significantly less expensive with $D \approx L$ from Eq. 8. For large number of picket robots, $n > 5$, the previously mentioned power-cycling or movement strategies can be employed to reduce the computational complexity to $D \approx L$ over a series of images rather than naively exhaustively searching a single image. In subsequent images where a bounding box has been found for each picket, the LED correspondence improves in performance up to $\frac{L!}{L-3!}$ times faster if the bounding boxes for all robots are completely non-overlapping. Non-overlapping bounding boxes occur when the robots are geographically spread out in the image. In multi-robot exploration, the intended use case of these team dynamic, the geographic spread is consistent such that the speedup is a achieved in practice. We evaluate the speedup in more detail Section 3.1.2.

From the P3P correspondence checks, Gauss-Newton minimization refines the initial solution from the P3P algorithm by minimizing reprojection error [20]:

$$P^* = \arg\min_P \sum_{<\mathbf{l},\mathbf{d}>\in\mathbf{C}} ||\pi(\mathbf{l}, P - \mathbf{d})||^2$$

where P is pose estimate, $\mathbf{l}$ is the set of LED configurations, $\mathbf{d}$ is the set of LED centroids, $\mathbf{C}$ is the LED correspondences, and π projects an LED from $\mathbb{R}^3$ into $\mathbb{R}^2$, the camera image.

In the update step of an EKF framework, the measurement model requires a covariance. We can directly calculate the pose estimate covariance, $\mathbf{Q}$, with the Jacobian ($\mathbf{J}$) from the Gauss-Newton minimization [20]:

$$\mathbf{Q} = (\mathbf{J}^T \Sigma^{-1} \mathbf{J})^{-1} \text{ where } \Sigma = \mathbf{I}_{2\times 2} \text{ pixels}^2 \tag{10}$$

2.3.2 Residual and Observation Matrix Definition

In this section we describe how the relative pose estimates from Section 2.3.1 are used to compute the EKF update. Following the block diagram from Figure 3, we have previously described the IMU and the pose estimation blocks. In addition, the propagation portion of the EKF block was covered in Section 2.2. For simplicity we assume that the measurements are in synchronous order until Section 2.5 such that we can

Figure 5: Example frame of relative pose estimation. Picket-1's pose tracker is in (a) and picket-2's pose tracker is in (b). Picket-1 and picket-2 share four colors respectively. The bounding boxes shown in blue correspond to the area where the pose estimator is searching for LEDS. LEDs associate are marked with a crosshair of their respective color on the associated pose estimator. The body frame of the picket robots are overlaid with the convention RGB-XYZ.

describe the sensor data buffer last. In order to perform an EKF update, we derive the residual and observation matrix that relates the relative pose estimates to the state vector as described in Section 2.1. The residual and the observation matrix are used to calculate the Kalman gain and correction in Section 2.4. Without the residual and observation matrices the relative pose estimates would serve without use.

An example overview of the multi-robot teams coordinates frames is shown in Figure 2. As a notation reminder, the camera frame is refered to as {C}, global frame as {G}, observer body as {O} and picket body {B}. The index i corresponding to the i^{th} picket robot.

A static camera transform is defined as:

$$[{}^{C}_{O}\bar{\mathbf{q}}^{T}, {}^{C}_{O}\mathbf{p}^{T}]^{T} \in \mathbb{R}^{7\times 1} \tag{11}$$

With respect to a single visible picket robot, i, a relative pose estimate from the camera frame on board the observer is defined as:

$$\mathbf{z}_i = [{}^{B}_{C}\bar{\mathbf{q}}^{T}_{i_z}, {}^{B}_{C}\mathbf{p}^{T}_{i_z}]^{T} \in \mathbb{R}^{7\times 1} \tag{12}$$

where the additional sub-subscript z refers to a observation in the EKF sense.

In an EKF framework, a residual, $\mathbf{r}$, and a measurement Jacobian, $\mathbf{H}$ are used to compute the EKF update. The standard relationship between the residual and measurement Jacobian is:

$$\mathbf{r} = \mathbf{z} - \hat{\mathbf{z}} \approx \mathbf{H}\tilde{\mathbf{x}} + \mathbf{n} \tag{13}$$

where $\mathbf{n}$ is noise. A prediction of the observation, $\hat{\mathbf{z}}_i$, is used to compute a residual in an EKF. This observation corresponds to a relative pose for each visible robot. Additionally, the quaternion states in $\mathbf{x}$ use the rotational error definition, $\delta\mathbf{q} = \mathbf{q}\otimes\hat{\mathbf{q}}^{-1}$ rather than the standard linear error, $\tilde{\mathbf{p}} = \mathbf{p} - \hat{\mathbf{p}}$. This follows from the error-state definition in Section 2.1.

To compute $\hat{\mathbf{z}}_i$, the state vector estimate is updated with the EKF propagation step. The poses of the picket robots are then converted from the global frame converted to the camera coordinate frame, {C}, in Eq. 11 to match the relative pose estimate. The rotation of the body frame, $\{B\}$, of the i^{th} picket robot with respect

to the global frame, $\{G\}$, is converted to the camera frame , $\{C\}$, by composing the camera and observer transforms. Similarly, the relative position between the camera and the i^{th} picket robot, ${}^{B}_{G}\hat{\mathbf{p}}_i - {}^{O}_{G}\hat{\mathbf{p}}_O - {}^{C}_{O}\mathbf{p}$, is converted to the camera frame. These coordinate frame transformations correspond to:

$$\hat{\mathbf{z}}_i = \begin{bmatrix} {}^{B}_{C}\hat{\mathbf{q}}_i \\ {}^{B}_{C}\hat{\mathbf{p}}_i \end{bmatrix} = \begin{bmatrix} {}^{B}_{G}\hat{\mathbf{q}}_i \otimes {}^{G}_{O}\hat{\mathbf{q}}_O \otimes {}^{O}_{C}\hat{\mathbf{q}}_O \\ {}^{C}_{O}\mathbf{R}{}^{O}_{G}\hat{\mathbf{R}}({}^{B}_{G}\hat{\mathbf{p}}_i - {}^{O}_{G}\hat{\mathbf{p}}_O - {}^{C}_{O}\mathbf{p}) \end{bmatrix} \tag{14}$$

where $\otimes$ represents quaternion multiplication. Note, rotation matrices can be computed directly from the corresponding quaternions. The definitions for quaternion multiplications and converting quaternions to rotation matrices are provided in Appendix A. From here onwards, rotation matrices and unit quaternions are used interchangeably.

The single-robot residuals with respect to each visible pickets robots are calculated according to the definition Eq. 13:

$$\mathbf{r}_i = \mathbf{z}_i - \hat{\mathbf{z}}_i = \begin{bmatrix} 2 \cdot \boldsymbol{\pi}({}^{B}_{C}\hat{\mathbf{q}}_i^{-1} \otimes {}^{B}_{C}\bar{\mathbf{q}}_{i_z}) \\ {}^{B}_{C}\mathbf{p}_{i_z} - {}^{B}_{C}\hat{\mathbf{p}}_i \end{bmatrix} = \begin{bmatrix} 2 \cdot \boldsymbol{\pi}(({}^{B}_{G}\hat{\mathbf{q}}_i \otimes {}^{G}_{O}\hat{\mathbf{q}}_O \otimes {}^{O}_{C}\hat{\mathbf{q}}_O)^{-1} \otimes {}^{B}_{C}\bar{\mathbf{q}}_{i_z}) \\ {}^{B}_{C}\mathbf{p}_{i_z} - {}^{C}_{O}\mathbf{R}{}^{O}_{G}\hat{\mathbf{R}}({}^{B}_{G}\hat{\mathbf{p}}_i - {}^{O}_{G}\hat{\mathbf{p}}_O - {}^{C}_{O}\mathbf{p}) \end{bmatrix} \tag{15}$$

where $\boldsymbol{\pi}$ is defined as $\boldsymbol{\pi}([\mathrm{q}_x, \mathrm{q}_y, \mathrm{q}_z, \mathrm{q}_w]^T)^T = [\mathrm{q}_x, \mathrm{q}_y, \mathrm{q}_z]^T$ and utilized as a small angle approximation for the orientation difference between $\mathbf{z}_i$ and $\hat{\mathbf{z}}_i$ as described in [18][19]. This reduces the dimensionality of residual, $\mathbf{r}_i \in \mathbb{R}^{6\times 1}$, from the measurement, $\mathbf{z}_i \in \mathbb{R}^{7\times 1}$ which is a requirement of the indirect EKF [18][19].

The i^{th} measurement Jacobian, $\mathbf{H}_i$, is calculated by applying small angle approximations and taking the partial derivatives of the i^{th} single-robot residual with respect to the error-state. The non-zero entries are shown below:

$$\begin{aligned} \mathbf{r}_i &\simeq \mathbf{H}_i\tilde{\mathbf{x}} \\ \mathbf{H}_i &= \begin{bmatrix} -{}^{C}_{G}\hat{\mathbf{R}} & \mathbf{0} & \mathbf{0} & \dots & {}^{C}_{G}\hat{\mathbf{R}} & \mathbf{0} & \mathbf{0} & \dots \\ {}^{C}_{G}\hat{\mathbf{R}}\lfloor({}^{B}_{G}\hat{\mathbf{p}}_i - {}^{O}_{G}\hat{\mathbf{p}}_O - {}^{C}_{O}\mathbf{p})\times\rfloor & -{}^{C}_{G}\hat{\mathbf{R}} & \mathbf{0} & \dots & \mathbf{0} & {}^{C}_{G}\hat{\mathbf{R}} & \mathbf{0} & \dots \end{bmatrix} \in \mathbb{R}^{6\times 16(n+1)} \\ \tilde{\mathbf{x}} &= \begin{bmatrix} {}^{G}\tilde{\boldsymbol{\theta}}_o & {}^{G}\tilde{\mathbf{p}}_o & {}^{G}\tilde{\mathbf{v}}_o & \dots & {}^{G}\tilde{\boldsymbol{\theta}}_i & {}^{G}\tilde{\mathbf{p}}_i & {}^{G}\tilde{\mathbf{v}}_i & \dots \end{bmatrix} \end{aligned} \tag{16}$$

where ${}^{C}_{G}\hat{\mathbf{R}} = {}^{C}_{O}\mathbf{R}{}^{O}_{G}\hat{\mathbf{R}}$ and $\lfloor\mathbf{q}\times\rfloor$ is the quaternion skew operator from Eq. 7. The higher order and cross terms are dropped from $\mathbf{H}_i$ to satisfy the linear requirement of the EKF.

The states of all picket robots become correlated with the observer robot through the measurement Jacobian. This enables an individual pose estimate of a picket robot to improve the state estimate of each picket robot. The correlation is essential to localizing the observer robot because it is unable to observe itself directly with the camera.

This correlation is essential because it is the reason a single stationary robot is able to mitigate drift IMU dead-reckoning drift. Starting with the first inchworm increment, the stationary robot functions as a SLAM landmark and has bounded covariance. Relative to the moving robots during any point of an increment, the stationary robot has the lowest uncertainty and can correct the IMU propagation error. This is especially critical in hazardous terrain where IMU propagation has increased noise. Each future inchworm increment can be viewed as composing the uncertainty of all the past increments.

This correlation builds in automatic loop-closures into the system. If a picket robot is left stationary for an extended period of time or indefinitely a future re-observation of the robot corrects position drift for all robots in the team. On a shorter time scale, when the observer loses sight of all pickets, potentially from failed pose estimation, errant path trajectory, or occlusion, a single re-observation of a stationary picket functions as a loop closure. This naturally follows from stationary robots states not being propagated so the covariance remains bounded and adds resiliency to the team.

2.4 EKF Update

From the camera measurement model the EKF update is performed.

The Kalman gain, $\mathbf{K}$, and covariance update are calculated using standard EKF equations:

$$\begin{aligned} \mathbf{K} &= \mathbf{\Sigma}_{k+1|k}\mathbf{H}^T(\mathbf{H}\mathbf{\Sigma}_{k+1|k}\mathbf{H}^T + \mathbf{Q})^{-1} \\ \mathbf{\Sigma}_{k+1|k+1} &= (\mathbf{I} - \mathbf{K}\mathbf{H})\mathbf{\Sigma}_{k+1|k} \end{aligned} \tag{17}$$

To utilize the standard equations, the overall measurement Jacobian is calculated by vertically stacking the single-robot measurement Jacobians from the camera measurement model in Eq. 16:

$$\mathbf{H} = [\mathbf{H}_1^T, \mathbf{H}_2^T, ... \mathbf{H}_{\text{n}}^T]^T \in \mathbb{R}^{6n \times 16(n+1)} \tag{18}$$

Accordingly the measurements, $\mathbf{z}_i$, are stacked identically:

$$\mathbf{z} = [\mathbf{z}_1^T, \mathbf{z}_2^T, ... \mathbf{z}_{\text{n}}^T]^T \in \mathbb{R}^{7n \times 1} \tag{19}$$

The corresponding overall observation noise is calculated by diagonalizing the uncorrelated relative pose estimate covariances from Eq. 10:

$$\mathbf{Q} = \text{diag}(\mathbf{Q}_1, \mathbf{Q}_2, ... \mathbf{Q}_n) \in \mathbb{R}^{6n \times 6n} \tag{20}$$

The correction for both observer and picket robots are calculated as:

$$\Delta\mathbf{x} = \mathbf{K}\mathbf{r} \tag{21}$$

For a single-robot, the correction has the form:

$$\begin{aligned} \Delta\mathbf{x}_i &= [\delta\boldsymbol{\theta}_i^T, \Delta\mathbf{p}_i^T, \Delta\mathbf{v}_i^T, \Delta\mathbf{b}_{i_g}^T, \Delta\mathbf{b}_{i_a}^T]^T \\ \Delta\mathbf{x} &= [\Delta\mathbf{x}_O^T, \Delta\mathbf{x}_1^T, ...\Delta\mathbf{x}_i^T]^T \end{aligned} \tag{22}$$

The non-quaternion states from the multi-robot state vector utilize standard additive correction. The orientation components of the correction are converted to quaternion corrections. Quaternion multiplication is applied to the quaternion states and corrections to complete the state update for the multi-robot team:

$$\begin{aligned} \hat{\mathbf{x}}_o &= \begin{bmatrix} {}^O_G\hat{\mathbf{q}} \otimes \delta\mathbf{q}_O \\ {}^G\hat{\mathbf{p}} + \Delta\mathbf{p}_O \\ {}^G\hat{\mathbf{v}} + \Delta\mathbf{v}_O \\ \hat{\mathbf{b}}_g + \Delta\mathbf{b}_{O_g} \\ \hat{\mathbf{b}}_a + \Delta\mathbf{b}_{O_a} \end{bmatrix}, \hat{\mathbf{x}}_i = \begin{bmatrix} {}^B_G\hat{\mathbf{q}} \otimes \delta\mathbf{q}_i \\ {}^G\hat{\mathbf{p}} + \Delta\mathbf{p}_i \\ {}^G\hat{\mathbf{v}} + \Delta\mathbf{v}_i \\ \hat{\mathbf{b}}_g + \Delta\mathbf{b}_{i_g} \\ \hat{\mathbf{b}}_a + \Delta\mathbf{b}_{i_a} \end{bmatrix} \\ \hat{\mathbf{x}} &= [\hat{\mathbf{x}}_O^T, \hat{\mathbf{x}}_1^T, \hat{\mathbf{x}}_2^T, ... \hat{\mathbf{x}}_i^T]^T \end{aligned} \tag{23}$$

2.5 Asynchronous System

Multi-robot systems have asynchronous data processing by nature. Kalman Filter approaches require measurements to be in synchronous order. A sensor data buffer that is emptied periodically is a viable solution to this problem. Shown in Algorithm 2 the buffer goes through a period of time collecting data and then processes the data on a constant period. When a IMU measurement or relative pose estimate is received, it is put into the sensor data buffer with insertion sort. Data that precedes the EKF's current time is discarded. The

buffer is emptied periodically at which point the data is processed synchronously according to Algorithm 1 and the buffer is cleared. The sensor data buffer is designed to handle the time delay between receiving IMU measurements from picket robots and the computation delay from receiving an image to producing relative pose estimates.

Period selection represents a tradeoff between discarding measurements and delays in feedback control. Shorter periods discard more measurements because the time and computation delay approaches the period. Longer periods delay feedback control as the EKF propagate and update are only computed when the buffer is emptied. In practice 10 Hz was used as the sensor data buffer frequency for a 3 robot team.

Algorithm 2: Asynchronous Sensor Data Buffer

```
while True, Rate = τ do
    wait()
    compute_messages()
    sensor_buffer.empty()
end
Function message_callback(message)
    if data.timestamp > filter_time then
        sensor_buffer.insertion_sort(data)
    end
Function compute_messages()
    foreach each measurement in sensor_buffer do
        if data.type == IMU then
            IMU_buffer.add(data.value)
            propagate_imu_state(IMU_buffer)
        else if data.type == RELATIVE_POSE then
            update_camera(data.value)
    end
```

The IMU model from Section 2.2 uses trapezoidal integration for the accelerometer and Runge Kutta for the gyroscope [18], [19] which both require the two most recent IMU messages. When the buffer is emptied and the measurements are processed, the two most recent IMU measurements for each robot are stored.

3 Results

3.1 Experimental Approach

We apply the localization technique described above to data collected from a team of three small, low-cost, mobile robots. The Zumy robot[1] is a decimeter-scale tracked robot running ROS on board a Linux computing system with networking and vision processing capability. The observer Zumy supports a Microsoft Lifecam 3000 webcam with 640× 480 pixels[2] at 30 Hz, InvenSense MPU-6050 MEMS IMU at 30 Hz, and supports WiFi wireless communication. This robot is designed to be easily built from commercially available off-the-shelf parts for a total cost of ≈ $350.

The robotic team consists of one observer and two picket robots shown in Figure 2. A Zumy robot with a camera serves as the observer, and to represent the inexpensive and less capable picket robots, we use Zumy robots without cameras. Each picket robot is outfitted with an LED "hat" so that it can be visually tracked by the observer robot. Infrared markers are attached to each Zumy for ground truth from a VICON motion capture system. The robots are manually driven.

The following subsections address the IMU characteristics, baseline performance of the relative pose estimator, and filter initialization procedure for the Zumy team.

3.1.1 IMU Characteristics

The IMU noise characteristics of the MPU 6050 [2] on board the Zumy are shown in Table 1 with a comparison to we used in the filter. The same noise parameter was used for all 3-axis of the gyroscope and accelerometer in the motion model. For example, $\mathbf{n}_g = [\sigma_g^2, \sigma_g^2, \sigma_g^2]^T$ and $\mathbf{n}_{wa} = [\sigma_{wa}^2, \sigma_{wa}^2, \sigma_{wa}^2]^T$. The parameters used in our filter were $\approx 2\times, 2.5\times$ the datasheet values to accomodate additional noise of having the IMU directly on the board without additional dampening. σ_{wg} and σ_{wa} are equivalent to those used on mobile devices [18] [19].

Table 1: IMU Parameters

	MPU 6050 Datasheet	Filter Values
$\sigma_g \left(\text{rad}/\text{s}\right)$	.050	.100
$\sigma_a \left(\text{m}/\text{s}^2\right)$	.0039	.0100
$\sigma_{wg} \left(\text{rad}/\text{s}^2\right)$	-	.0001
$\sigma_{wa} \left(\text{m}/\text{s}^3\right)$	-	.0001

3.1.2 Relative Pose Estimation

One of the primary requirements of relative pose estimation is the ability to work in a variety of lighting conditions. LEDs are very bright relative to the background of images. False detection rates and computation time can be improved by setting the webcam exposure to low because of the higher contrast between the LEDs and background of the image. Camera images from datasets with and without lighting are barely distinguishable as demonstrated by Figure 6. The performance of the relative pose estimators is identical under both lighting conditions. Pose measurements were dropped past the range of 3m. For 5 LEDs, there is a 60 times theoretical improvement in LED correspondence step as shown in Eq. 9. In practice, this causes an approximately 5 times speedup in the multi-color approach as compared to the single-color approach.

The general LED layout of the baseplate on picket-1 is shown in Figure 7 and dimensions with respect to the center of the baseplate in Table 2. Our setup has a slightly lower baseline, lower resolution camera and less

[1] `https://wiki.eecs.berkeley.edu/biomimetics/Main/Zumy`

[2] `https://www.invensense.com/products/motion-tracking/6-axis/mpu-6050/`

Figure 6: Example image from a dataset with the lights off in (a). Example image from a dataset with lights on in (b). Lighting condition does not have a noticeable effect on the pose estimator performance.

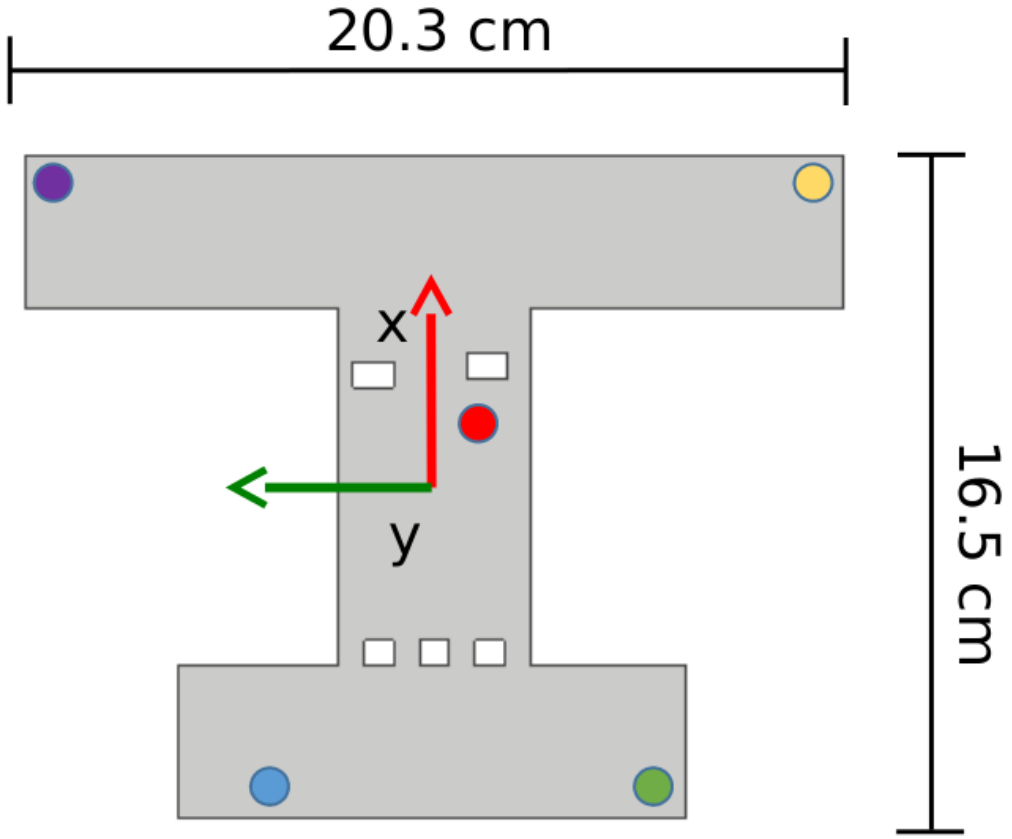

Figure 7: LED layout for picket-1 shown in Figure 2. The colored circles on the baseplate correspond to the appropriate color LED. A summary of the exact dimensions relative to the baseplate is in Table 2.

Table 2: LED Placement Picket-1

	Purple LED	Yellow LED	Red LED	Blue LED	Green LED
x (cm)	7.9	7.5	1.7	-7.4	-7.4
y (cm)	8.7	-9.6	-0.9	4.9	-5.5
z (cm)	7.8	6.9	9.8	5.1	2.2

accurate LED centroids from the difference in RGB vs infared LEDs. A direct comparison between the system setup in [20] and our system setup is in Table 3.

For the purposes of our experiments, most inchworm increments were done at pose depth of approximately 2m. This means that when the observer starts moving, the stationary picket robot is approximately 2m away. At a distance of 2m away the pose tracker position and orientation errors were approximately 5-8cm and 1-4 degrees respectively. The higher uncertainty range as compared to [20] is due to having non-uniform baselines in each axis. This causes variations in performance depending on pose geometry between the observer and picket.

3.1.3 Filter Initialization

The multi-robot team initialization was done with a "burn in" approach with the robots at rest. The observer is defined to start at the origin unless specified otherwise. The velocity of each robot is assumed to be identically zero for the specified time. The IMU measurements for each robot are averaged. This initializes the bias of the gyroscope and accelerometer in the state vector in Eq. 1. The initial pose of the picket robots relative to the observer are found by averaging relative pose estimates. The initial covariance of the bias is given by the Table 1 and the uncertainty of the pose is computed during pose estimation in Eq. 10.

3.2 Planar Base Case

The baseline experimental task was a cooperative U-turn in planar 3-DOF with one observer and two pickets. The robots were manually driven in the dark. Although the data set was recorded in a 3-DOF environment, the filter was not constrained with environmental priors.

In Figures 8 and 9, we show the resulting trajectories of this planar U-turn set as compared to ground truth. We plot the results of using only one picket while discarding the measurements from other, and then the results of using both pickets. Note that the observer trajectory is not as smooth as that of picket-1 or picket-2, because the motion of the observer has un-modeled vibration effects that cause motion blur and temporary changes to the "static" camera transform.

This data set consisted of 10 inchworm increments. End-position drift for using a single picket vs. using two pickets are shown in Table 4. Fusion refers to the camera and IMU approach described in Section2. Fusion one picket in Table 4 refers to the case where the filter is run only with the observer and picket-1. In this case, picket-2 does not exist and the corresponding values in the table are left blank. The other case, the filter is run on picket-2 and observer is omitted because it does not meet the stationary robot requirement of the inchworm strategy. The angular drift, which causes error in the future, for the two picket fusion case was less than the one picket case. The position drift was slightly less for the two picket case, but otherwise consistent with the one picket case. Although unconstrained to a plane, the angular drift was almost exclusively in yaw. Performing right or left turns with the robot team introduces more rotational drift than forward or backwards motions. One source of rotational error is latent biases in the system. The second source of rotational error is correlated errors in pose estimates when only picket-1 was visible and picket-2 was experiencing temporary line of sight

Table 3: Pose Tracker Comparison

	Faessler et al. [20]	Our System
Resolution (pixels^2)	752x480	640x480
Baseline Radius (cm)	10.9	10.6
LEDs/Robot	5	5
LED Type	Infared	RGB
$\approx$ error at 2 m depth	5 cm, 1-2 deg	5-8 cm, 1-4 deg

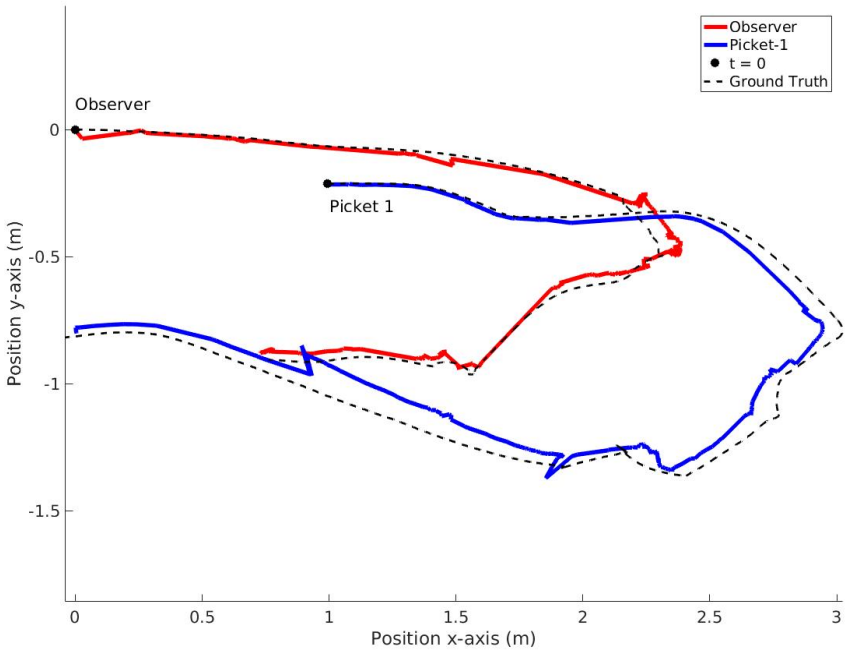

Figure 8: A plot of the XY projection of the team's pose estimates from the EKF along with the ground truth trajectories. Shown for the base case of the planar U-turn where a single picket is used to perform localization.

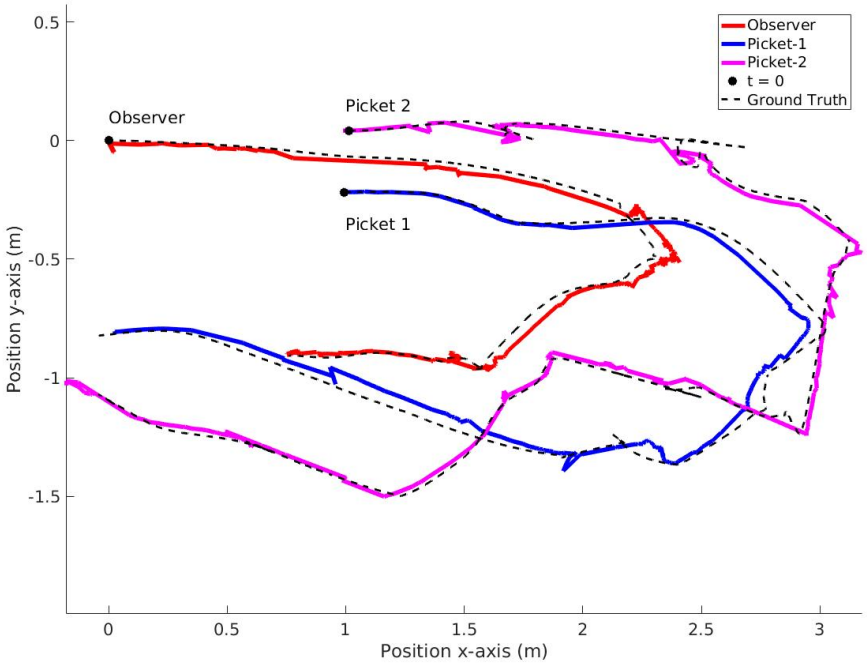

Figure 9: A plot of the XY projection of the team's pose estimates from the EKF along with the ground truth trajectories. Shown for the base case of the planar U-turn where both pickets are used to perform localization.

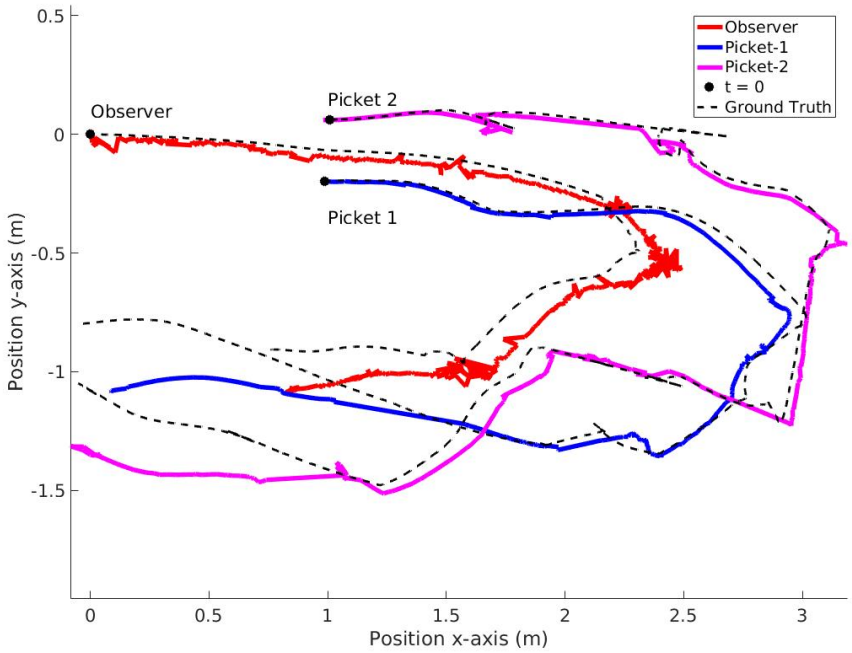

Figure 10: Camera only approach: A plot of the XY projection of the team's pose estimates along with the ground truth trajectories, using a camera only approach. Performs notably worse in yaw drift than the IMU-camera fusion approach shown in Figures 8, 9.

Table 4: Planar Drift Analysis

		Camera Only Two Pickets	Fusion One Picket	Fusion Two Pickets
Observer	x (cm)	-8.14	**0.67**	-0.80
	y (cm)	16.64	2.58	**-1.46**
	z (cm)	4.09	-5.42	**3.07**
	Angle (°)	9.33	1.89	**1.54**
Picket-1	x (cm)	-13.38	**-4.25**	-6.23
	y (cm)	28.97	-1.79	**-1.70**
	z (cm)	4.53	-7.91	**4.39**
	Angle (°)	4.72	2.64	**1.36**
Picket-2	x (cm)	**-1.35**	-	9.92
	y (cm)	25.40	-	**-5.65**
	z (cm)	4.56	-	**5.18**
	Angle (°)	4.56	-	**1.88**

Table 5: Line of Sight (LoS) Failure Analysis

	Fusion One Picket	Fusion Two Pickets
Picket-1 (% images in LoS)	91.6	91.6
Picket-2 (% images in LoS)	-	57.5
max LoS failure (sec)	38.10	9.28

(LoS) failure. Without external features or global correction, the yaw errors persist until the end of experiment, but adding more picket robots helps to mitigate these effects. The jagged regions of the trajectory correspond

to the observer and picket robots starting or stopping motion. The LED mount and the robots shake during these transient motions.

A camera only filtering approach was evaluated in Figure 10 as a baseline and it performed significantly worse with four times as much yaw drift than the IMU plus camera fusion approach. The camera only approach uses the same formulation of the measurement derived in Section 2.3.2 but without a motion model. Without the gyroscope, the inchworm localization performs significantly worse in orientation estimation. Using IMU for dead-reckoning produced drifts of over 100 m for each of the 3 robots. Additionally, removing the stationary constraint and always propagating the IMU state, produced errors of over 30 m for each of the 3 robots. The inchworm stationary constraint is essential because the team operates without global correction.

LoS failure occurs when a picket robot is not observed in a camera image regardless of cause. LoS failure occurs most commonly when a picket robot leaves the field of view of the camera, errant observer trajectories, partial LED occlusions, or on rare occasions the pose estimator fails to converge. A LoS failure analysis is shown in Table 5. Note the camera only and fusion two pickets have the same line of sight failure characteristics. The LoS of Picket-1 was prioritized when manually driving the robots. Picket-2 was frequently out of LoS only during the turning sections. As noted by the maximum continuous time of LoS failure times, the filter is able to provide accurate localization during prolonged LoS failure because at least a single robot remains stationary throughout the duration.

3.3 Non-planar Terrain with Ramp

The second experiment was conducted in an environment featuring non-planar terrain, obstacles, and occlusions. The robots were manually driven in the environment shown in Figure 11. The 6-DOF planar data set consisted of 10 inchworm increments: 3 for the rock garden and 7 for the right turn and ramp. Temporary line of sight failure of both pickets occurred during the rock garden because the pose estimator failed to converge, as the observer was moving on the rocks. Wheel slippage occurred during the rock garden section. After the rock garden, picket-2 was deliberately left behind to simulate a hole in the environment and a loss of a robot.

The ground truth trajectories and the EKF pose estimates of the data set are shown in Figures 13 and 14. The end point drift analysis is shown in Table 6 with a comparison against a camera only approach. The fusion approach outperformed the camera only for the observer and picket-1. The most critical improvement is the orientation error of the observer which persists without correction. Picket-2 traveled mostly in a straight line except during the rock garden, and the endpoint errors of both approaches are almost identical. The drift is predominately in pitch for each robot. Temporal plots with ground truth are in Figures 15, 16,17 and 18. The observer's trajectory is visually less smooth because of motion blur and vibration of the camera, non-static transform. The EKF position and orientation estimate of the observer stabilizes when it becomes stationary. The orientation and position drift about each leapfrog increment are similar to random walk processes.

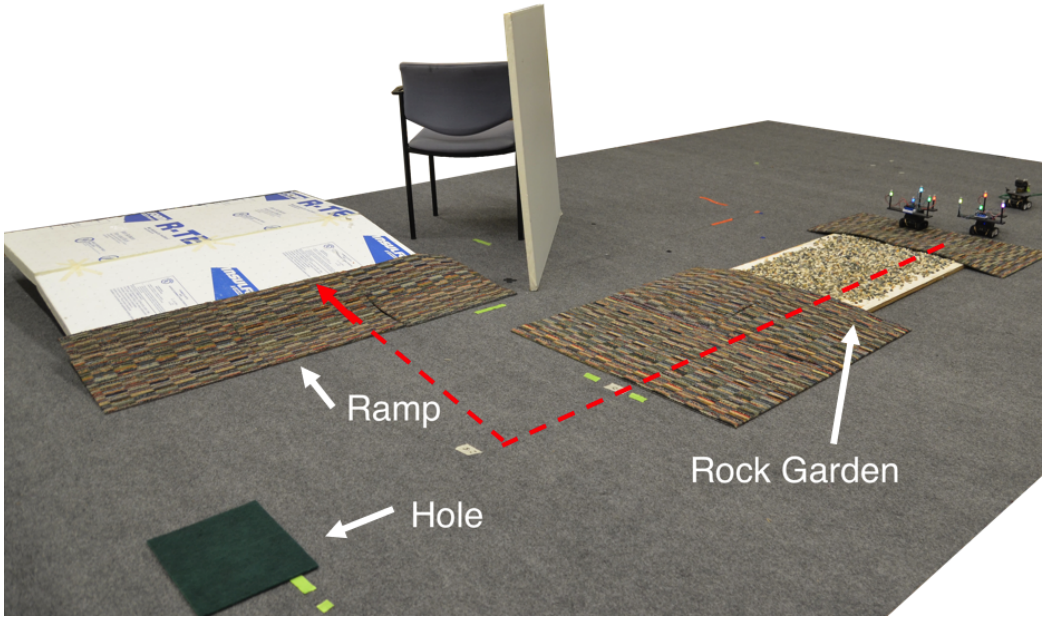

Figure 11: 6-DOF environment for testing: Robot team is on the right, rock garden is center-right), the green patch on the bottom-left is a "hole", and the ramp is on top left.

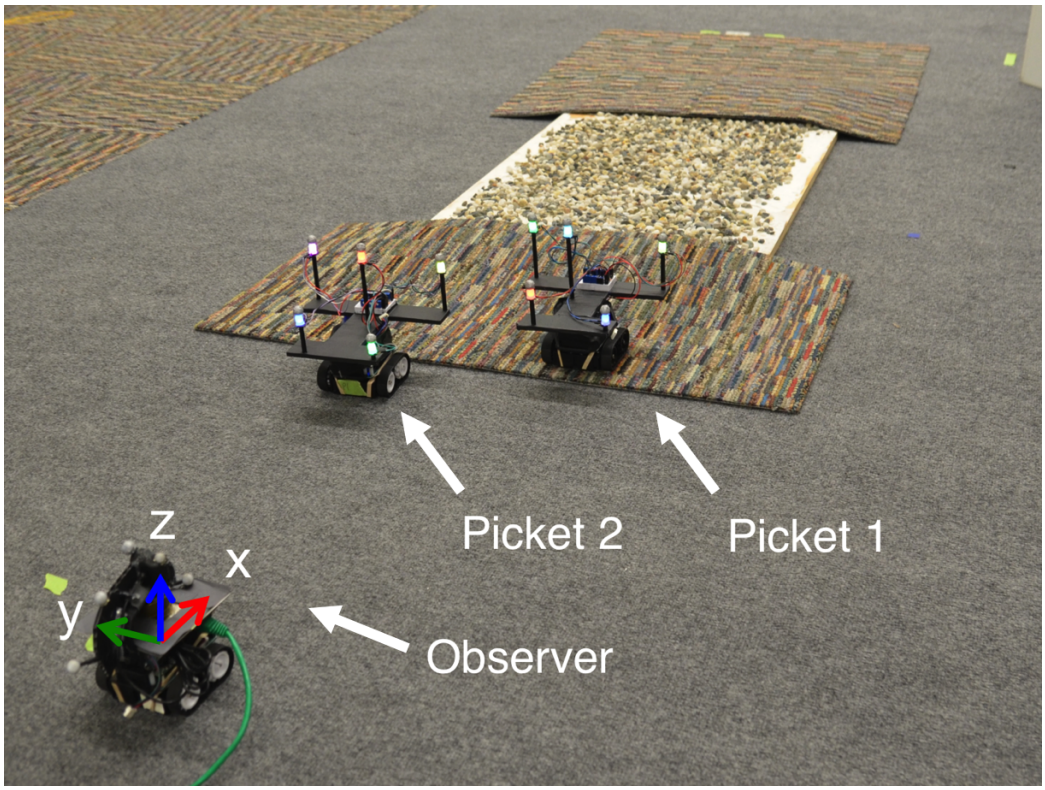

Figure 12: Starting position of the robot team with view of the rock garden section. The origin is defined as the starting position of the observer.

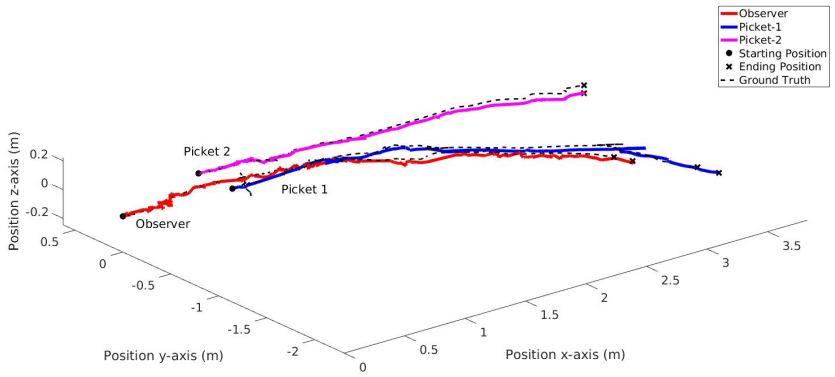

Figure 13: Ground truth trajectories of the multi-robot team are compared against the estimates of the EKF for the non-planar environment. Axes are scaled equally

.

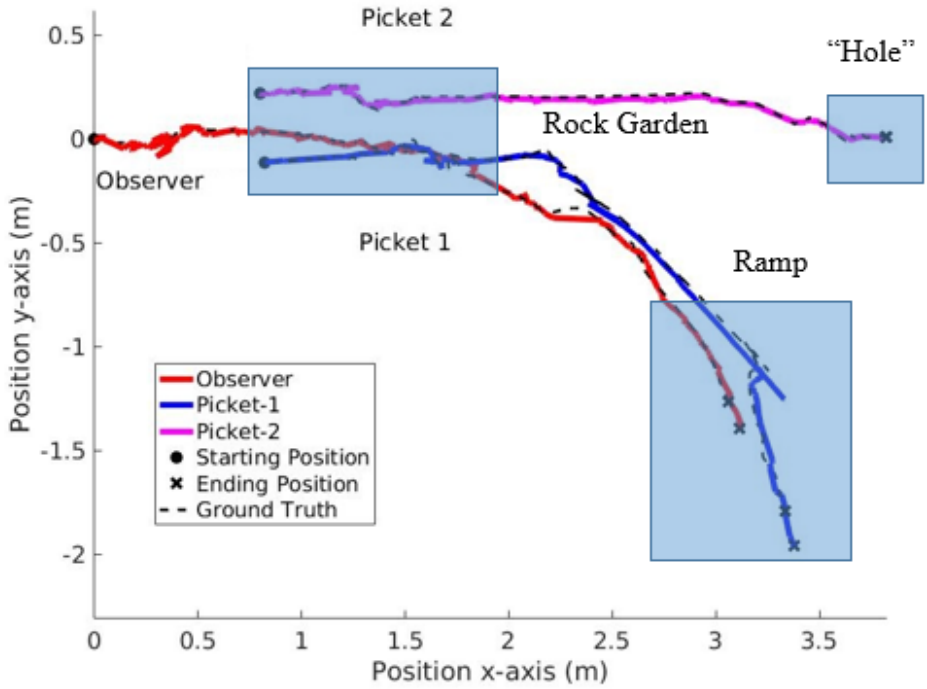

Figure 14: 2D projection of ground truth trajectories of the multi-robot team are compared against the estimates of the EKF for the non-planar environment.

Table 6: Non-Planar Drift Analysis

		Camera Only Two Pickets	Fusion Two Pickets
Observer	x (cm)	**-4.03**	-5.35
	y (cm)	13.36	**12.67**
	z (cm)	1.01	**0.04**
	Angle (°)	3.12	**2.18**
Picket-1	x (cm)	**-2.11**	-4.48
	y (cm)	17.9	**16.76**
	z (cm)	1.72	**0.10**
	Angle (°)	6.22	**4.29**
Picket-2	x (cm)	**0.28**	0.32
	y (cm)	-0.20	**0.15**
	z (cm)	**5.37**	**5.37**
	Angle (°)	**3.58**	3.59

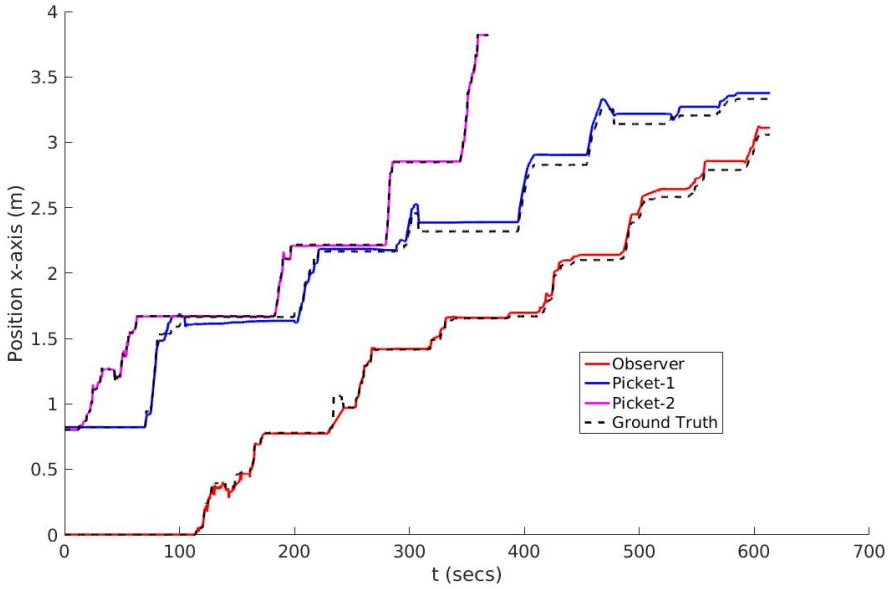

Figure 15: Position along the x-axis versus time.

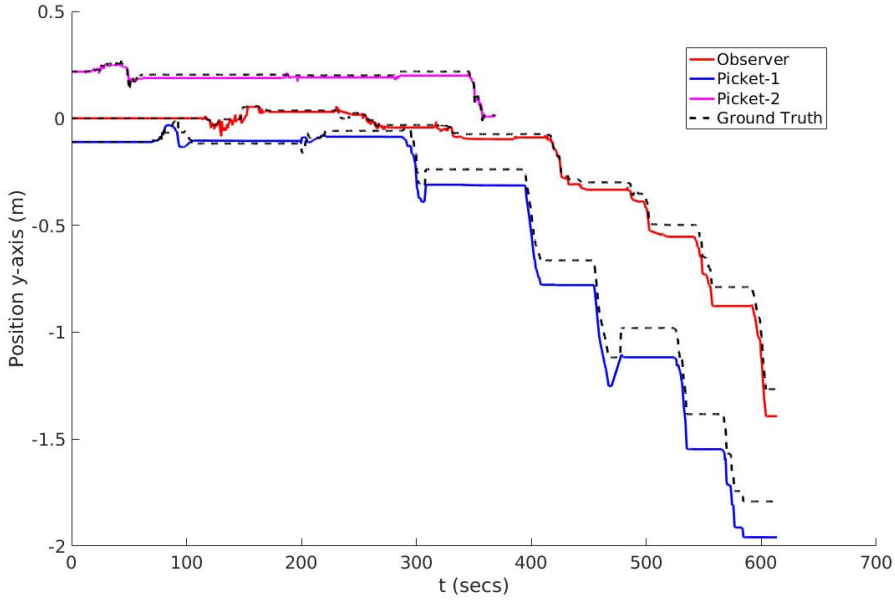

Figure 16: Position along the y-axis versus time.

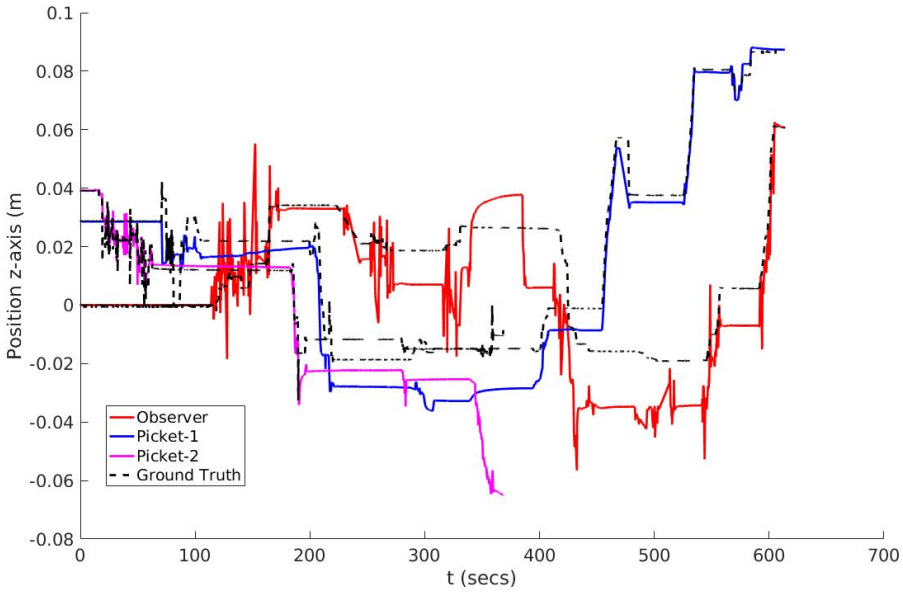

Figure 17: Position along the z-axis versus time.

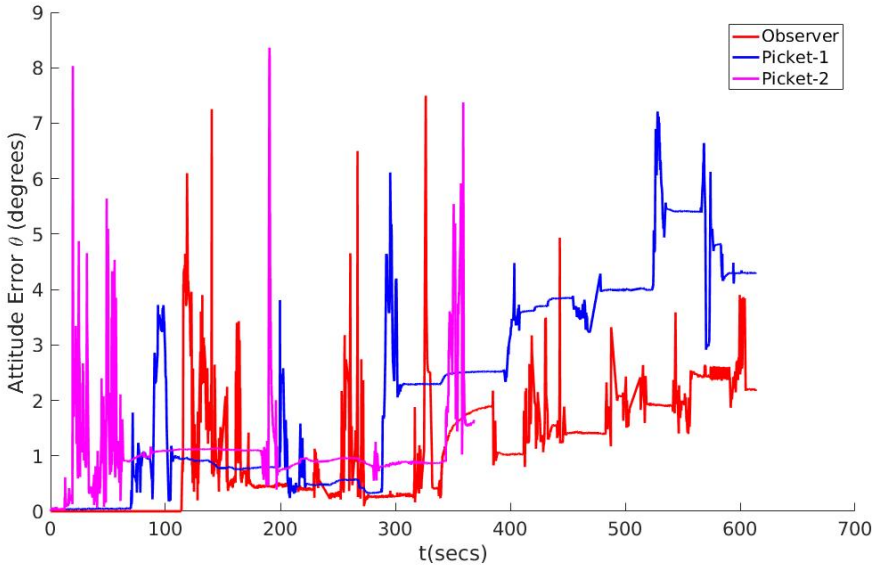

Figure 18: Orientation error versus time.

4 Conclusion

A heterogeneous team which consists of a single "observer" and multiple "picket" robots is able to navigate a visually featureless, unknown, non-planar environment as a unit, using only relative pose observations and IMU measurements to estimate the motion of the team. The IMU and camera fusion approach presented in this thesis has clear advantages over a simpler camera only approach that outweigh the cost of having asynchronous communication. This follows from Figures 8, 9, and 10. Although calibration of an IMU adds many complications, it is a natural choice for environments in which wheel encoders are unreliable. A camera-only approach heavily relies on line of sight at all times, which is restricting and potentially very hard to maintain. In addition, motion-model based approaches for EKF propagation allow for the rejection of errant camera pose estimates from faulty LED detections or P3P correspondence matching. Most importantly, with a camera only approach, each inchworm increment consisting of picket and observer movement has an associated positional and rotational drift in 6-DOF. The calibration of the IMU allows the fusion based approach of using the stationary robots' gravity vectors to reduce and bound the pitch and roll drift leading to drift in only 4-DOF.

In the future, we will develop exploration strategies for larger robot teams where of more than 10 robots. Design considerations in exploration include motions that mitigate yaw drift and team formations for effectively passing over hazards. Autonomous control with increasing number of robots is far more effective than manual driving. Simple proportional-integral-derivate (PID) position controllers are insufficient in dealing with obstacles because the robots are easily flipped upside down or on their sides. A more advanced control scheme that factors in collaboration of robots is needed for effective exploration in hazardous environments.

Another direction of future interest is multi-team exploration. Picket robots are designed to be cheap and disposable by nature. They can be left stationary for an indefinite amount of time to create permanent landmarks. Permanent landmarks are very effective at creating loop closures and finding when independently exploring teams path cross. However, by leaving a picket robot behind a team operates with less capabilities. The long term behavior of picket robots, whether leaving them behind as landmarks or keeping them with their respective team, is an open optimization problem.

References

[1] Robert Grabowski, Luis E Navarro-Serment, Christiaan JJ Paredis, and Pradeep K Khosla. Heterogeneous teams of modular robots for mapping and exploration. *Autonomous Robots*, 8(3):293–308, 2000.

[2] Duncan W Haldane, Péter Fankhauser, Roland Siegwart, and Ronald S Fearing. Detection of slippery terrain with a heterogeneous team of legged robots. In *IEEE Int. Conf. on Robotics and Automation*, pages 4576–4581, 2014.

[3] Luis E Navarro-Serment, Christiaan JJ Paredis, Pradeep K Khosla, et al. A beacon system for the localization of distributed robotic teams. In *Int. Conf. on Field and Service Robotics*, volume 6, 1999.

[4] Stephen Tully, George Kantor, and Howie Choset. Leap-frog path design for multi-robot cooperative localization. In *Int. Conf. on Field and Service Robotics*, pages 307–317. Springer, 2010.

[5] Ryo Kurazume and Shigeo Hirose. An experimental study of a cooperative positioning system. *Autonomous Robots*, 8(1):43–52, 2000.

[6] Karl E Wenzel, Andreas Masselli, and Andreas Zell. Visual tracking and following of a quadrocopter by another quadrocopter. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4993–4998. IEEE, 2012.

[7] Thumeera R Wanasinghe, George KI Mann, and Raymond G Gosine. Distributed collaborative localization for a heterogeneous multi-robot system. In *27th IEEE Canadian Conf. on Electrical and Computer Engineering (CCECE), 2014*, pages 1–6.

[8] Thumeera R Wanasinghe, George KI Mann, and Raymond G Gosine. Distributed leader-assistive localization method for a heterogeneous multirobotic system. *IEEE Transactions on Automation Science and Engineering*, 12(3):795–809, 2015.

[9] Francesco Mondada, Edoardo Franzi, and Paolo Ienne. Mobile robot miniaturisation: A tool for investigation in control algorithms. pages 501–513. Experimental Robotics III, Springer Berlin Heidelberg, 1994.

[10] Manuela Veloso, Peter Stone, Kwun Han, and Sorin Achim. The cmunited-97 small robot team. pages 242–256. Springer Berlin Heidelberg, 1998.

[11] FR Fabresse, F Caballero, and A Ollero. Decentralized simultaneous localization and mapping for multiple aerial vehicles using range-only sensors. In *IEEE Int. Conf. on Robotics and Automation, 2015*, pages 6408–6414.

[12] Kostas E Bekris, Max Glick, and Lydia E Kavraki. Evaluation of algorithms for bearing-only slam. In *IEEE Int. Conf. on Robotics and Automation, 2006*, pages 1937–1943.

[13] Luca Carlone, Miguel Kaouk Ng, Jingjing Du, Basilio Bona, and Marina Indri. Rao-Blackwellized particle filters multi robot SLAM with unknown initial correspondences and limited communication. In *IEEE Int. Conf. on Robotics and Automation, 2010*, pages 243–249.

[14] Nikolas Trawny, Stergios I Roumeliotis, and Georgios B Giannakis. Cooperative multi-robot localization under communication constraints. In *IEEE Int. Conf. on Robotics and Automation, 2009*, pages 4394–4400.

[15] James McLurkin. Using cooperative robots for explosive ordnance disposal. pages 1–10, Massachusetts Institute of Technology Artificial Intelligence Laboratory, 1996.

[16] Raj Madhavan, Kingsley Fregene, and Lynne E Parker. Distributed heterogeneous outdoor multi-robot localization. In *IEEE Int. Conf. on Robotics and Automation, 2002*, volume 1, pages 374–381.

[17] Dennis Strelow and Sanjiv Singh. Motion estimation from image and inertial measurements. *The International Journal of Robotics Research*, 23(12):1157–1195, 2004.

[18] Anastasios I Mourikis and Stergios I Roumeliotis. A multi-state constraint Kalman filter for vision-aided inertial navigation. In *IEEE Int. Conf. on Robotics and Automation, 2007*, pages 3565–3572.

[19] Guoquan Huang, Michael Kaess, and John J Leonard. Towards consistent visual-inertial navigation. In *IEEE Int. Conf. on Robotics and Automation, 2014*, pages 4926–4933.

[20] Matthias Faessler, Elias Mueggler, Karl Schwabe, and Davide Scaramuzza. A monocular pose estimation system based on infrared LEDs. In *IEEE Int. Conf. on Robotics and Automation, 2014*, pages 907–913.

[21] Andreas Breitenmoser, Laurent Kneip, and Roland Siegwart. A monocular vision-based system for 6D relative robot localization. In *IEEE Int. Conf. on Intelligent Robots and Systems, 2011*, pages 79–85.

[22] Long Quan and Zhongdan Lan. Linear n-point camera pose determination. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(8):774–780, 1999.

[23] Laurent Kneip, Davide Scaramuzza, and Roland Siegwart. A novel parametrization of the perspective-three-point problem for a direct computation of absolute camera position and orientation. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2011*, pages 2969–2976.

[24] Aamir Ahmad, Gian Diego Tipaldi, Pedro Lima, and Wolfram Burgard. Cooperative robot localization and target tracking based on least squares minimization. In *IEEE Int. Conf. on Robotics and Automation, 2013*, pages 5696–5701.

[25] Vadim Indelman, Erik Nelson, Nathan Michael, and Frank Dellaert. Multi-robot pose graph localization and data association from unknown initial relative poses via expectation maximization. In *IEEE Int. Conf. on Robotics and Automation, 2014*, pages 593–600.

[26] Abdulmuttalib T Rashid, Mattia Frasca, Abduladhem A Ali, Alessandro Rizzo, and Luigi Fortuna. Multi-robot localization and orientation estimation using robotic cluster matching algorithm. *Robotics and Autonomous Systems*, 63:108–121, 2015.

[27] Esha D Nerurkar, Stergios I Roumeliotis, and Agostino Martinelli. Distributed maximum a posteriori estimation for multi-robot cooperative localization. In *IEEE Int. Conf. on Robotics and Automation, 2009*, pages 1402–1409.

[28] Stergios I Roumeliotis and George A Bekey. Distributed multirobot localization. *IEEE Transactions on Robotics and Automation*, 18(5):781–795, 2002.

[29] Agostino Martinelli, Frederic Pont, and Roland Siegwart. Multi-robot localization using relative observations. In *IEEE Int. Conf. on Robotics and Automation, 2005*, pages 2797–2802.

[30] Andrew Howard. Multi-robot simultaneous localization and mapping using particle filters. *The International Journal of Robotics Research*, 25(12):1243–1256, 2006.

[31] Amanda Prorok, Alexander Bahr, and Alcherio Martinoli. Low-cost collaborative localization for large-scale multi-robot systems. In *IEEE Int. Conf. on Robotics and Automation, 2012*, pages 4236–4241.

[32] Nikolas Trawny and Stergios I Roumeliotis. Indirect Kalman filter for 3d pose estimation. *University of Minnesota, Dept. of Comp. Sci. & Eng., Tech. Rep*, 2, 2005.

A JPL Quaternion Reference

In this section we define the quaternion, unit quaternion, quaternion functions, and the small angle linearization used for indirect EKF propagation and update steps [18] [19] [32].

Quaternion definition:

$$\mathbf{q} = \begin{bmatrix} \mathrm{q}_x \\ \mathrm{q}_y \\ \mathrm{q}_z \\ \mathrm{q}_w \end{bmatrix}$$

Unit quaternion definition:

$$\bar{\mathbf{q}} = \frac{\mathbf{q}}{||\mathbf{q}||} = \begin{bmatrix} k_x \sin(\theta/2) \\ k_y \sin(\theta/2) \\ k_z \sin(\theta/2) \\ \cos(\theta/2) \end{bmatrix} = \begin{bmatrix} \boldsymbol{k} \sin(\theta/2) \\ \cos(\theta/2) \end{bmatrix}$$

Quaternion Inverse:

$$\bar{\mathbf{q}}^{-1} = \begin{bmatrix} -\mathrm{q}_x \\ -\mathrm{q}_y \\ -\mathrm{q}_z \\ \mathrm{q}_w \end{bmatrix}$$

Quaternion product:

$$\mathbf{q} \otimes \mathbf{p} = \begin{bmatrix} \mathrm{q}_w\mathrm{p}_x + \mathrm{q}_z\mathrm{p}_y - \mathrm{q}_y\mathrm{p}_z + \mathrm{q}_x\mathrm{p}_w \\ -\mathrm{q}_z\mathrm{p}_x + \mathrm{q}_w\mathrm{p}_y + \mathrm{q}_x\mathrm{p}_z + \mathrm{q}_y\mathrm{p}_w \\ \mathrm{q}_y\mathrm{p}_x - \mathrm{q}_x\mathrm{p}_y + \mathrm{q}_w\mathrm{p}_z + \mathrm{q}_z\mathrm{p}_w \\ -\mathrm{q}_x\mathrm{p}_x - \mathrm{q}_y\mathrm{p}_y - \mathrm{q}_z\mathrm{p}_z + \mathrm{q}_w\mathrm{p}_w \end{bmatrix}$$

Skew operator:

$$\lfloor \mathbf{q} \times \rfloor = \begin{bmatrix} 0 & -\mathrm{q}_z & \mathrm{q}_y \\ \mathrm{q}_z & 0 & -\mathrm{q}_x \\ -\mathrm{q}_y & \mathrm{q}_x & 0 \end{bmatrix}$$

Cross product:

$$\mathbf{q} \times \mathbf{p} = \lfloor \mathbf{q} \times \rfloor \mathbf{p} = \begin{bmatrix} \mathrm{q}_y\mathrm{p}_z - \mathrm{q}_z\mathrm{p}_y \\ \mathrm{q}_z\mathrm{p}_x - \mathrm{q}_x\mathrm{p}_z \\ \mathrm{q}_x\mathrm{p}_y - \mathrm{q}_y\mathrm{p}_x \end{bmatrix}$$

Convert quaternion to rotation matrix:

$$\mathbf{R} = \mathbf{C}(\bar{\mathbf{q}}) = \begin{bmatrix} \mathrm{q}_x^2 - \mathrm{q}_y^2 - \mathrm{q}_z^2 + \mathrm{q}_w^2 & 2(\mathrm{q}_x\mathrm{q}_y + \mathrm{q}_z\mathrm{q}_w) & 2(\mathrm{q}_x\mathrm{q}_z - \mathrm{q}_y\mathrm{q}_w) \\ 2(\mathrm{q}_x\mathrm{q}_y - \mathrm{q}_z\mathrm{q}_w) & -\mathrm{q}_x^2 + \mathrm{q}_y^2 - \mathrm{q}_z^2 + \mathrm{q}_w^2 & 2(\mathrm{q}_y\mathrm{q}_z + \mathrm{q}_x\mathrm{q}_w) \\ 2(\mathrm{q}_x\mathrm{q}_z + \mathrm{q}_y\mathrm{q}_w) & 2(\mathrm{q}_y\mathrm{q}_z - \mathrm{q}_x\mathrm{q}_w) & -\mathrm{q}_x^2 - \mathrm{q}_y^2 + \mathrm{q}_z^2 + \mathrm{q}_w^2 \end{bmatrix}$$

Small angle linearization:
A small angular error, $\delta\bar{\mathbf{q}}$, in axis angle form can be approximated with the following:

$$\delta\bar{\mathbf{q}} = \begin{bmatrix} \boldsymbol{k}\sin(\tilde{\theta}/2) \\ \cos(\tilde{\theta}/2) \end{bmatrix} \approx \begin{bmatrix} \frac{1}{2}\tilde{\boldsymbol{\theta}} \\ 1 \end{bmatrix}$$

Equivalently in rotation matrix form:

$$\mathbf{R} = \mathbf{C}(\delta\bar{\mathbf{q}}) \approx \mathbf{I}_3 - \lfloor\tilde{\boldsymbol{\theta}}\times\rfloor$$

The linearization stems from $\cos(x) \approx 1, \sin(x) \approx x$ for small x.

B EKF Reference

In this section we list the standard prediction and update equations for an Extended Kalman Filter.

Prediction:

$\hat{\mathbf{x}}_{k|k-1} = f(\hat{\mathbf{x}}_{k-1|k-1}, \mathbf{u}_k)$
$\mathbf{\Sigma}_{k|k-1} = \mathbf{F}_{k-1}\mathbf{\Sigma}_{k-1|k-1}\mathbf{F}_{k-1}^T + \mathbf{S}_k$

Update:

$\mathbf{K}_k = \mathbf{\Sigma}_{k|k-1}\mathbf{H}_k^T(\mathbf{H}_k\mathbf{\Sigma}_{k|k-1}\mathbf{H}_k^T + \mathbf{Q}_k)^{-1}$
$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k(\mathbf{z}_k - h(\hat{\mathbf{x}}_{k|k-1}))$
$\mathbf{\Sigma}_{k|k} = (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\mathbf{\Sigma}_{k|k-1}$

Definitions:

$\hat{\mathbf{x}}_k =$ state vector
$\mathbf{\Sigma}_k =$covariance matrix
$\mathbf{z} =$ measurement vector
$\mathbf{u} =$ control input
$f =$ state transition model
$h =$ observation model
$\mathbf{F} = \left.\dfrac{\partial f}{\partial \mathbf{x}}\right|_{\hat{\mathbf{x}}_{k-1|k-1}, \mathbf{u}_k} =$ Jacobian of state transition model
$\mathbf{H}_k = \left.\dfrac{\partial h}{\partial \mathbf{x}}\right|_{\hat{\mathbf{x}}_{k|k-1}} =$ Jacobian of observation model
$\mathbf{K} =$ Kalman gain
$\mathbf{Q} =$ observation noise covariance
$\mathbf{S} =$ process noise covariance