Heterogeneous Sensor Fusion for Accurate State Estimation of Dynamic Legged Robots

Simona Nobili[1*], Marco Camurri[2*], Victor Barasuol[2], Michele Focchi[2],
Darwin G. Caldwell[2], Claudio Semini[2] and Maurice Fallon[3]

[1] School of Informatics,
University of Edinburgh,
Edinburgh, UK
simona.nobili@ed.ac.uk

[2] Advanced Robotics Department,
Istituto Italiano di Tecnologia,
Genoa, Italy
{firstname.lastname}@iit.it

[3] Oxford Robotics Institute,
University of Oxford,
Oxford, UK
mfallon@robots.ox.ac.uk

***Abstract*—In this paper we present a system for the state estimation of a dynamically walking and trotting quadruped. The approach fuses four heterogeneous sensor sources (inertial, kinematic, stereo vision and LIDAR) to maintain an accurate and consistent estimate of the robot's base link velocity and position in the presence of disturbances such as slips and missteps. We demonstrate the performance of our system, which is robust to changes in the structure and lighting of the environment, as well as the terrain over which the robot crosses. Our approach builds upon a modular inertial-driven Extended Kalman Filter which incorporates a rugged, probabilistic leg odometry component with additional inputs from stereo visual odometry and LIDAR registration. The simultaneous use of both stereo vision and LIDAR helps combat operational issues which occur in real applications. To the best of our knowledge, this paper is the first to discuss the complexity of consistent estimation of pose and velocity states, as well as the fusion of multiple exteroceptive signal sources at largely different frequencies and latencies, in a manner which is acceptable for a quadruped's feedback controller. A substantial experimental evaluation demonstrates the robustness and accuracy of our system, achieving continuously accurate localization and drift per distance traveled below $1\,\mathrm{cm/m}$.**

I. Introduction

For legged robots to be useful and eventually autonomous, they must be able to reliably walk and trot over a variety of terrains and in the presence of disturbances such as slips or pushes. They must also be able to perceive their environment and to avoid collisions with obstacles and people.

Legged robot control systems typically act to regulate the position, the orientation, and the associated velocities of the robot's base or center of mass. This state vector is used for the planning of body trajectories, balancing and push recovery, as well as local mapping and navigation. Accurate and reliable state estimation is essential to achieve these capabilities, but it is a challenging problem due to the demands of low latency and consistency that high-frequency feedback control place on it. Meanwhile, impulsive ground impacts, aggressive turns and sensor limitations cause many modern exteroceptive navigation algorithms to fail when most needed.

Despite the improvements demonstrated by bipedal systems in the DARPA Robotics Challenge, for example [12], quadruped robots (Boston Dynamics LS3 [14], MIT Cheetah 2 [18], ANYmal [11]) present a more immediate solution to explore the parts of the world that are inaccessible to traditional robots.

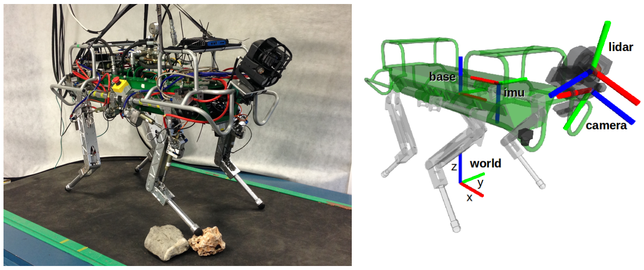


Fig. 1: *Left:* the Hydraulic Quadruped robot (HyQ) with the Carnegie Robotics Multisense SL sensor head at the front. *Right:* the main coordinate frames used in this paper.

In this paper, we demonstrate how inertial, kinematic, stereo vision and LIDAR sensing can be combined to produce a modular, low-latency and high-frequency state estimate which can be directly used to control a state-of-the-art dynamic quadruped. In turn, this estimate can be used to build accurate maps of the robot's environment and to enable navigational autonomy. Compared with prior research, this contribution is the first to discuss the complexity of consistent estimation of both position and velocity signals, as well as the fusion of stereo vision and LIDAR at very different sensor frequencies and latencies for a quadruped's feedback controller.

This article is presented as follows: in Section II we describe previous research in this field and discuss how our contribution differs from it. The robot platform and its exteroceptive sensor suite are described in Section III, as are the performance requirements we wish to achieve. Our algorithmic contribution is presented in Section V, first overviewing our core inertial-kinematic state estimator before describing modules for stereo odometry and LIDAR registration. In particular, we discuss the appropriate manner in which these sources of information should be fused into the main estimate. In Section VII, we present experimental results where we show how each of the sensor modalities behaves in challenging circumstances and how they contribute to improved performance. In the results section, the robot is demonstrated achieving continuously accurate localization with drift per distance traveled below $1\,\mathrm{cm/m}$.

II. Related Work

There is a significant body of literature in state estimation and navigation of legged robots. As Ma et al. [14] described,

[*] Both authors contributed equally to this work.

performance can be distinguished by multiple factors, such as the quality of the sensors, the dynamics of the robot's motion, as well as the degree of challenge of the test environments and extensiveness of the testing performed. To that list we would add the quality of velocity estimation and suitability for use in closed loop control.

Exteroceptive and proprioceptive state estimation are often dealt with differently. Exteroceptive state estimate is closely related to Simultaneous Localization and Mapping (SLAM) and Barfoot [2] is an excellent resource in this area.

The motivation for proprioceptive state estimation is somewhat different for legged control system. Notably, Blösch et al. [3] presented a rigorous treatment of the fusion of leg kinematics and IMU information with a particular focus on estimator consistency, which becomes important when fusing very different signal modalities.

The method of sensor fusion we present is similar to that of Chilian et al. [6] which discussed stereo, inertial and kinematic fusion on a six-legged crawling robot measuring just $35\,\mathrm{cm}$ across – yet combining all the required sensing on board. It was unclear if computation was carried out on-board. The work of Chitta et al. [7] is novel in that it explored localization against a known terrain model using only contact information derived from kinematics.

With a focus on *perception in the loop*, the electrically-actuated MIT Cheetah 2 [18] produces impressive jumping gaits which are cued off of a LIDAR obstacle detection system. Because their work focuses on control and planning, the perception system used therein is not intended to be general nor it is used for state estimation.

The work of Ma et al. [14] is most closely related to ours in scale and dynamism of their robot. Their system was designed to function as a modular sensor head fusing a tactical grade inertial measurement unit with stereo visual odometry to produce a pose estimate for navigation tasks such as path planning. Robot's kinematic sensing was only used when visual odometry failed. Their approach was focused on pose estimation and was not used within the robot's closed loop controller. Their extensive evaluation (over thousands of meters) achieved 1% error per distance traveled.

For cost and practical reasons we wish to avoid using such high quality inertial sensors where possible. Our approach was developed with a MEMS IMU in mind. In all of our experiments we recorded both MEMS and Fiber Optic IMUs. In Section VII we present some initial results comparing the performance when using either sensor.

Finally, the estimator used in this work is based on a loosely-coupled EKF. This general approach has been applied to micro-aerial flight including Shen et al. [21] and Lynen et al. [13].

III. Experimental Scenario

Our experimental platform is a torque-controlled Hydraulic Quadruped robot (HyQ, Figure 1) [20]. The system is $1\,\mathrm{m}$ long, and weighs approximately $85\,\mathrm{kg}$. Its 12 revolute joints have a rotational range of $120°$. A summary of the core sensors on the robot is provided in Table I. The $1\,\mathrm{kHz}$ sensors are read by our control computer (using a real-time operating system). All other sensors are connected to a perception computer and are passively synchronized with the real-time sensors [17].

Sensor	Sensor Freq.	Sensor Latency	Integration Freq.	Integration Latency	Variables Measured
IMU	1000	< 1	1000	n\a	${}_{b}\boldsymbol{\omega}_{b}$ ${}_{b}\ddot{\mathbf{x}}_{b}$
Joint Encoders	1000	< 1	1000	< 1	${}_{b}\dot{\mathbf{x}}_{b}$
LIDAR	40	10	0.2-0.5	600	${}_{w}\mathbf{x}_{b}$ ${}_{w}\boldsymbol{\theta}_{b}$
Stereo	10	125	10	42	${}_{w}\mathbf{x}_{b}$

TABLE I: Frequency (Hz) and latency (ms) of the main sensors and for computing corresponding filter measurements.

The robot's main exteroceptive sensor is the Carnegie Robotics Multisense SL which is composed of a stereo camera and a Hokuyo UTM-30LX-EW planar ranging laser (LIDAR). The laser produces 40 line scans per second with $30\,\mathrm{m}$ maximum range — while spinning about the forward-facing axis. Every few seconds, it spins half a revolution and a full 3D point cloud is accumulated. The stereo camera was configured to capture 1024×1024 images at $10\,\mathrm{Hz}$ and has a $0.07\,\mathrm{m}$ baseline. Within the unit, a Field Programmable Gate Array (FPGA) carries out Semi-Global Matching (SGM) [9] to produce a depth image from the image pair. The depth image is used to estimate the depth of point features in Section V-B as well as for other terrain mapping tasks. Figure 2 shows an example of a left camera image and a depth image taken during an experiment — indicating the challenging scenarios we target.

IV. Requirements

The purpose of the state estimator is to produce a low drift estimate of the floating base of the robot model, which is typically the main link of the robot with the IMU rigidly attached to it. The estimate should have low latency (including transduction and data transmission) which is important for the velocity estimates used in the feedback loop of a controller. Low drift or drift-free state estimation is also used in navigation tasks (such as mapping and trajectory planning) as basic building block for many autonomous systems.

Our system is designed such that our core estimator requires only inertial and kinematic measurements to achieve low drift (with varying drift rates for different gaits). The additional sensing modalities of stereo vision and LIDAR can be incorporated in a manner which is complementary and provides

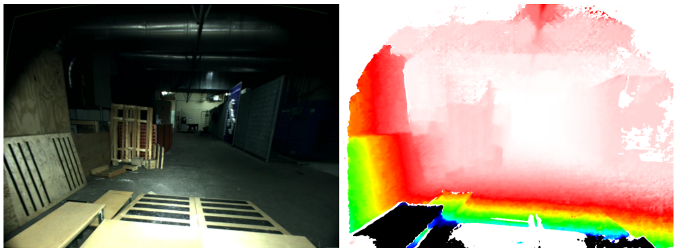


Fig. 2: Example of left camera image and depth image produced by the robot's stereo camera. This reflects the difficult lighting conditions and challenging structure of the test arena. The scene is illuminated with the sensor's on-board lights.

redundancy to mechanical compliance and deformation in the terrain (*e.g.,* mud or loose stones). As the exteroceptive sensors are captured with much lower frequency and higher latency (Table I), care must be taken in how their inputs are incorporated into the estimate.

V. Approach

We build upon an inertial-kinematic estimator recently described in [5]. In this section, we overview the core approach and use the same notation introduced therein. The 15 elements of the robot's base link state vector are defined by:

$$\mathcal{X} = \begin{bmatrix} {}_w\mathbf{x}_b & {}_b\dot{\mathbf{x}}_b & {}_w\boldsymbol{\theta}_b & \mathbf{b}_a & \mathbf{b}_\omega \end{bmatrix} \tag{1}$$

where the base velocity ${}_b\dot{\mathbf{x}}_b$, is expressed in the base frame b, while the position ${}_w\mathbf{x}_b$ and orientation ${}_w\boldsymbol{\theta}_b$ are expressed in a fixed world frame w (the list of frames and their location on HyQ is depicted in Figure 1). The orientation is expressed as quaternion, but the attitude uncertainty is tracked by the exponential coordinates of the perturbation rotation vector, as described in [4]. The state vector is completed by IMU acceleration and angular velocity biases $\mathbf{b}_a$ and $\mathbf{b}_\omega$, which is updated by an EKF from [8].

Measurements of acceleration and angular velocity are taken from the IMU at $1\,\mathrm{kHz}$. These are transformed into the base frame (subject to the estimated biases) to estimate the base acceleration ${}_b\ddot{\mathbf{x}}_b$ and angular velocity ${}_b\boldsymbol{\omega}_b$. Then, EKF is propagated using a direct inertial process model.

IMU biases are typically estimated when the robot is stationary and held static thereafter, as they are difficult to infer on a dynamic robot[1]. When operating, the robot drift of the yaw estimate is a significant issue. We have typically used a Microstrain 3DM-GX4-25 IMU but more recently explored using the KVH 1775, a tactical grade IMU equipped with a Fiber Optic Gyroscope (FOG). For this reason, we compare the estimation performance of both IMUs in Section VII.

A. Leg Odometry Module

Joint sensing contributes through a Leg Odometry (LO) module [5], which also runs at $1\,\mathrm{kHz}$. During the filter update step, a measure for the base velocity ${}_b\dot{\mathbf{x}}_b$ is computed as a combination of the individual velocity measurements ${}_b\dot{\mathbf{x}}_{b_f}$ from each in-stance foot f, as follows:

$${}_b\dot{\mathbf{x}}_{b_f} = -{}_b\dot{\mathbf{x}}_f - {}_b\boldsymbol{\omega}_b \times {}_b\mathbf{x}_f, \tag{2}$$

where ${}_b\dot{\mathbf{x}}_f$ and ${}_b\mathbf{x}_f$ are the velocity and position of foot f in the base frame, respectively.

As the robot is not equipped with contact sensors, we use the probabilistic contact classifier described in [5] to infer the combination of feet which are in stable and reliable contact. The velocity measure is then a weighted combination of the individual components, proportional to the probability of a particular foot being in reliable contact. An adaptive covariance associated with the velocity measurement accounts for harsh impact forces (up to $600\,\mathrm{N}$ when trotting) and helps ensure a smooth and accurate velocity estimate.

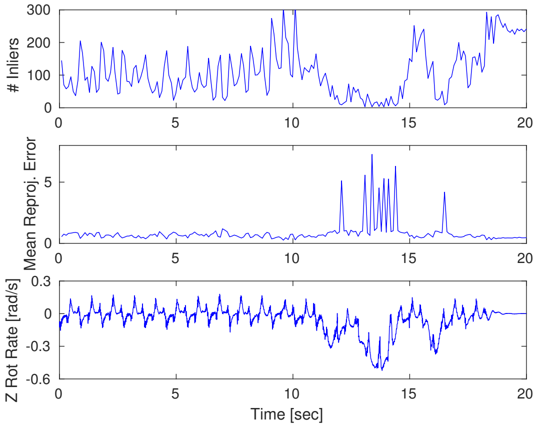


Fig. 3: Visual odometry performance during a trotting sequence: the robot first trots forward at $0.3\,\mathrm{m/s}$ and then turns in place sharply over a $5\,\mathrm{s}$ period. During the initial trotting phase, VO performance is satisfactory. However, image blur causes the number of inliers to fall and mean re-projection error to spike. During this part of the experiment, no VO measurement packets are incorporated into the main motion estimate.

In experiments with trotting and crawling gaits, the proprioceptive estimator achieved drift rates of approximately $3\,\mathrm{cm}$ per meter traveled. This (or greater) accuracy is needed to build accurate terrain maps (in motion) and to allow the robot's rear feet to achieve desired footholds (of 2–3 cm size) when sensed by the robot's forward facing sensors.

B. Visual Odometry Module

Visual Odometry (VO), and more broadly Visual SLAM, is becoming more feasible on legged platforms. This is enabled by more rugged sensors which are less susceptible to failure due to the dynamic motion of the robot. Nonetheless, certain types of robot motions (in particular ground impacts and aggressive turns) cause motion blur, especially in low light conditions.

Chilian et al. [6] suggested that leg odometry and visual odometry can be complementary, as difficult terrain often contains texture. In our experience however, where locomotion struggles (such as with a mis-timed footstep) it instead *induces* motion blur and reduces VO performance (Figure 3). Latency is another important issue to consider. As stated in [14], a camera packet is typically received once every 50 inertial measurement packets.

Our visual odometry pipeline uses the open source implementation of FOVIS [10]. While its performance is competitive with more recent approaches, it could be straightforwardly replaced by a more recent VO system such as ORB-SLAM [15]. Its only input is a sequence of left/depth image pairs. It tracks FAST features in a key-frame approach so as to estimate incremental camera motion, from image frame $k-1$ to frame k which we denote ${}_c\hat{T}_c^{\,k-1:k}$, where c indicates the camera frame. Using the known camera-to-body frame transformation, ${}_bT_c$, this can be expressed at the corresponding estimate of the motion of

[1] In [14] the robot was commanded to stand still occasionally to back out rotation rate bias estimates.

the body frame from $k-1$ to k as:

$$ {}_b\hat{T}_b^{\,k-1:k} = {}_bT_c \; {}_c\hat{T}_c^{\,k-1:k} ({}_bT_c)^{-1} \tag{3} $$

We have considered a number of ways of incorporating this information into the $1\,\mathrm{kHz}$ running estimate. The manner in which it is incorporated can conflict with other signal sources. Due to the accuracy of the gyroscope sensor, we currently incorporate only the translation element and as a result that orientation estimate can drift in yaw.

Velocity measurement: Initially we explored using the VO signal as a second velocity source. Operating the camera at its highest frequency ($30\,\mathrm{Hz}$), a measure of velocity can be computed by differencing the incremental motion estimate

$$ {}_b\hat{\dot{\mathbf{x}}}_b = \frac{{}_b\hat{\mathbf{x}}_b^{\,k-1:k}}{(t_k - t_{k-1})} \tag{4} $$

where t_k is the timestamp of image frame k. While this signal does approximate velocity, this is unsatisfactory because of the low frequency and high latency of the camera.

Frame-to-frame position measurement: A more straightforward approach is to use this relative motion estimate to infer a position measurement of the robot relative to a previous state of the filter.

Taking the posterior estimate of the EKF filter corresponding to time t_{k-1}, a measurement of the pose of the body at time t_k can be computed as follows:

$$ {}_w\hat{T}_b^{\,k} = {}_wT_b^{\,k-1} \oplus {}_b\hat{T}_b^{\,k-1:k} \tag{5} $$

This can be incorporated as an EKF position measurement. Ma et al. [14] used this approach to estimate the robot pose estimate (at $7.5\,\mathrm{Hz}$) and occasionally relied on LO when a failed VO frame occurred. Probabilistic fusion of redundant signal sources was not carried out. Instead, our goal is consistent estimation of position and velocity at high frequency, which makes subtleties of the integration important.

Consider Figure 4, which shows the position estimate of the robot while trotting. Overlaid on the figure are red markers indicating the timestamps of image frames. Any pose estimate computed using VO would be below the Nyquist frequency of the robot's motion and demand very precise time synchronization.

Position measurement over several frames: We choose a less fragile approach which integrates the visual motion estimate over several image frames and to compute a compounded EKF position measurement. Specifically, we integrate the VO estimate for a N-frame window ${}_b\hat{T}_b^{\,k-N:k}$ to form a position measurement in the world frame as follows:

$$ {}_w\hat{T}_b^{\,k} = {}_wT_b^{\,k-N} \oplus {}_b\hat{T}_b^{\,k-N:k} \tag{6} $$

where N is the number of frames used for integration (typically N corresponded to 2–3 s). This is similar to key-frame tracking where tracking for an extended duration can improve accuracy over frame-to-frame tracking. Finally the position portion of this measurement, ${}_w\hat{\mathbf{x}}_b^{\,k}$, is then used to create an EKF correction of the body position.

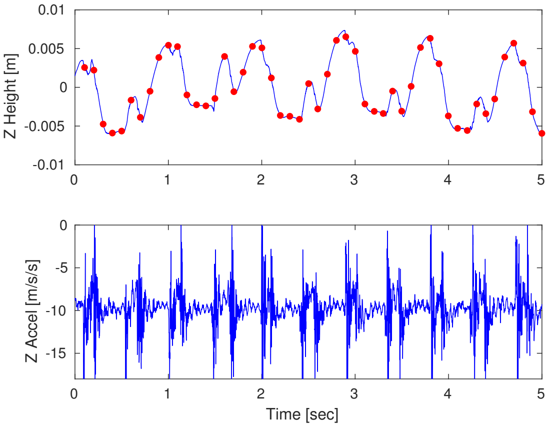


Fig. 4: Height (z-dimension) of the robot's base frame (top) and raw z-axis accelerometer measurements (bottom) while trotting. Indicated with red dots are timestamps of received stereo camera images. The bandwidth of the base motion is much higher than for many wheeled robots, while foot strikes cause acceleration spikes.

C. *LIDAR-based Localization*

To incorporate information from the LIDAR sensor, we use Iterative Closest Point (ICP) registration of 3D point clouds to estimate the robot's pose. Using the terminology of [19], this involves aligning a *reading* cloud to a *reference* cloud so as to infer the relative position of the sensor which captured the clouds. In particular, we want to measure (at time k) the relative pose ${}_w\hat{T}_b^{\,k}$ between the robot's base frame b and the world frame w, and then incorporate it as an observation in the EKF.

Registration of consecutive point clouds is often used to incrementally estimate motion, but it accumulates error over time. On the other hand, repeatedly registering to a common reference cloud is difficult when the robot moves away from its original position, as the overlap between the reference and the current cloud decreases over time.

In [16], we proposed a strategy for non-incremental 3D scene registration, which shows increased robustness to initial alignment error and variation in overlap. That work extended the *libpointmatcher* ICP implementation of Pomerleau et al. [19] with pre-filtering of the input clouds and automatic tuning of the outlier-rejection filter to account for the degree of point cloud overlap. The approach, called Auto-tuned ICP (AICP), leverages our low drift inertial-kinematic state estimate to initialize the alignment (Section V-A) and to compute an overlap parameter $\Omega \in [0,1]$ which can tune the filter. The parameter is a function of the maximum range and the field of view of the LIDAR sensor.

Here, we use the AICP framework to prevent accumulated drift and maintain accurate global localization. In our experiments, we could reliably register point clouds with only 11% overlap, which corresponded to a position offset of approximately $13\,\mathrm{m}$.

Forced reference update: When the overlap drops dramatically, a reference point cloud update is required. In this work, we extend the AICP algorithm to trigger a reference point

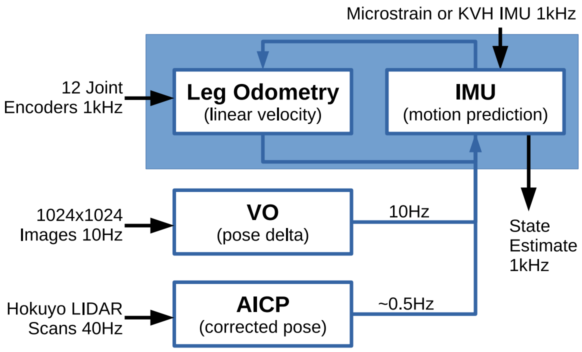


Fig. 5: State estimator signal block diagram: the core inertial-kinematic block (blue) runs on the control computer, while the other modules run on the perception computer.

cloud update when Ω decreases below the empirical threshold of 11%. When the threshold is crossed, the reference cloud is updated with the most recent reading cloud, whose alignment was successful. We follow three heuristics to determine if an alignment is successful. First, the mean residual point-wise error should be smaller than the threshold α:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n} r_i < \alpha \tag{7}$$

where $r_1, \ldots, r_n$ are the residual distances between the accepted matching points in the input clouds. Second, the median of the residual distribution, $Q(50)$, should be smaller than the threshold α:

$$Q(50) < \alpha \tag{8}$$

Third, the quantile corresponding to the overlap measure should be also smaller than α:

$$Q(\Omega) < \alpha \tag{9}$$

The first two conditions are commonly used metrics of robustness, while the third automatically adapts to the degree of point cloud overlap. The parameter α was set to $0.01\,\mathrm{m}$ during our experiments.

The limited frequency of the Hokuyo ($40\,\mathrm{Hz}$) and the speed of rotation of the sensor define the density of the accumulated point clouds. Increasing the spin rate reduces the density of each cloud. When trotting at $0.5\,\mathrm{m/s}$, a sensor spin rate of $15\,\mathrm{RPM}$ corresponds to a new point cloud every $2\,\mathrm{s}$ — with the robot traveling about a body length in that time. Running on a parallel thread, the AICP algorithm produces a pose correction with a computation time of approximately $600\,\mathrm{ms}$.

VI. Implementation Design

Filtering a heterogeneous set of signals with different frequencies and latencies requires careful consideration. A block diagram of our system is presented in Figure 5, with timing information for acquisition and integration in Table I.

At each iteration of the main $1\,\mathrm{kHz}$ inertial-kinematic loop, we calculate the prediction step and then immediately output the predicted state estimate ($\mathcal{X}^k$) to the control system to minimize latency. Subsequently, a velocity measurement is calculated using the $1\,\mathrm{kHz}$ leg odometry. This is applied to the filter as a Kalman update. These two components run in a single thread with no inter-process communication between them.

The visual odometry and the LIDAR registration modules operate at much lower frequencies and higher latencies. The VO pipeline takes no input other than the camera imagery and outputs the relative distance estimate at $10\,\mathrm{Hz}$. The acquisition time for our stereo camera is significant ($125\,\mathrm{ms}$) — partially due to the SGM algorithm [9] (running on the FPGA) and image transport.

The LIDAR scans are received with much lower latency, but are then accumulated into a point cloud before the registration algorithm computes an alignment. The corrected pose estimate is then calculated and transmitted to the core estimator in the same manner as for VO — albeit at much lower frequency. Thus, both modules run as decoupled processes without affecting the core estimator.

Considerations due to latency: The implementation of the filter maintains a history of measurements so as to enable asynchronous corrections with significant latency — specifically the VO and LIDAR corrections. In Figure 6, we explain how this filter works with a toy example (for simplicity, leg odometry is left out of this discussion). In blue is the best estimate of the state over the history at that moment in time. In red is the effect of EKF update steps caused by measurements. In green are portions of the filter history which have been overwritten due to a received measurement.

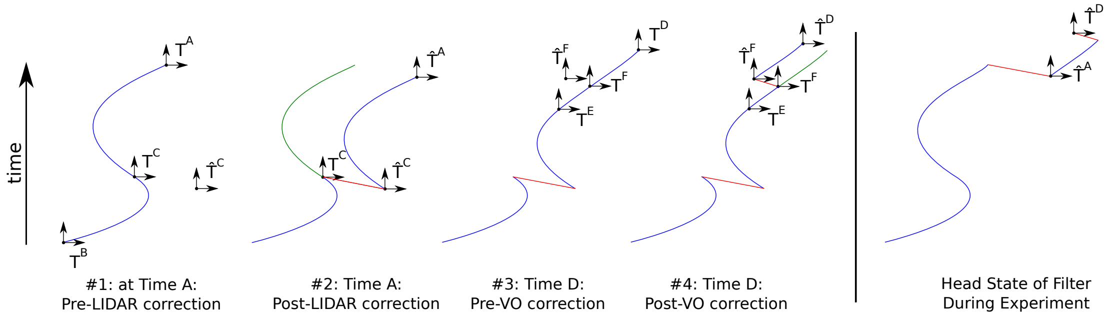
Event #1: Before Event #1, the IMU process model will have been predicting the state of the robot until Time A. At this instant, a LIDAR correction is received which is based on LIDAR line scans collected over a period of several seconds stretching from Time B to Time C. This means that the position correction estimate from the LIDAR over that period is significantly delayed when it is finally computed. Also the accumulation is dependent on the accurate IMU+LO state estimate — which creates a coupling between these modules.

Event #2: The LIDAR measurement is incorporated as an EKF correction which produces the posterior estimate $\hat{T}^C$ which causes the mean of the EKF to shift. The remaining portion of the state estimate is recalculated to incorporate the correction (such that the head of the filter becomes $\hat{T}^A$). The green trajectory is overwritten (this is a crucial step).

Event #3: Over the next period of time the filter continues to predict the head of the estimator using the IMU process model. At Time D, a new visual odometry measurement is created which measures the relative transformation of the body frame between Time E and Time F as ${}_b\hat{T}_b{}^{E:F}$. This measurement is typically received with about $170\,\mathrm{ms}$ of delay.

Event #4: We wish to use this information to correct the pose of the robot towards $\hat{T}^F$, as described in Section V-B. The key step is that this correction to the filter is carried out using the re-filtered trajectory (mentioned in Event #2). After the correction is applied, the head of the filter becomes $\hat{T}^D$ and the estimator continues as normal.

The final sub-figure (on the right) shows the state of the head of the filter over the course of the example. This is the running



#1: at Time A: Pre-LIDAR correction

#2: Time A: Post-LIDAR correction

#3: Time D: Pre-VO correction

#4: Time D: Post-VO correction

Head State of Filter During Experiment

Fig. 6: Example illustrating how VO and LIDAR measurements can be incorporated into the filter despite having much higher latency than the IMU process model. In blue is the best estimate of the trajectory at that instance, in green are parts of the trajectory which have been recomputed after a position measurement was incorporated (in red). Elapsed time is indicated in the upward direction. All indicated coordinate frames are of the base frame expressed in the world frame.

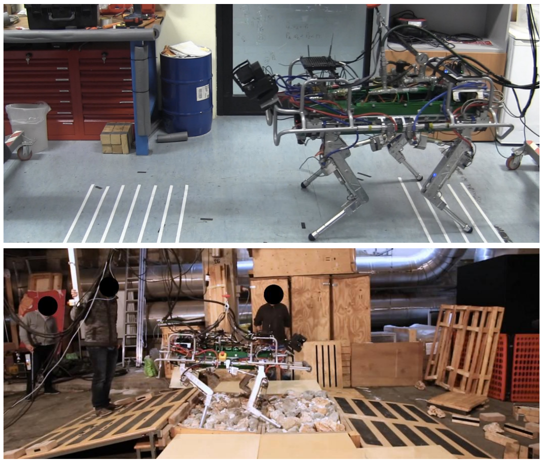


Fig. 7: Environments used to test repeatability (top) and to compare algorithm variants in challenging scenarios (bottom).

estimate that would have been available to the controller online.

VII. Experimental Results

To validate the described system, we performed experiments in two different scenarios. First, to demonstrate accuracy and repeatability, a repetitive experiment was carried out in a laboratory environment using a Vicon motion capture system to generate the ground truth.

Second, extensive testing was carried out in a poorly lit, industrial area with a feature-less concrete floor, as well as test ramps and rock beds (Figure 7, bottom). The environment, the different locomotion gaits (trotting and crawling) and the uneven terrains presented a large number of challenges to our algorithms and demonstrated the importance of using redundant and heterogeneous sensing. The robot's peak velocity when trotting was about $0.5\,\mathrm{m/s}$, which is approximately half of typical human walking speed.

We will refer to four different configurations: the baseline inertial-kinematic estimator (IMU-LO) and three variants which use either VO, AICP, or both. Except where noted, we used the KVH 1775 IMU in our experiments.

A video to accompany this paper is available at https://youtu.be/39Y1Jx1DMO8

A. *Experiment 1: Validation and Repeatability*

The robot was commanded to continuously trot forward and backward to reach a fixed target (a particular line in Figure 7, top). Robot position and velocity estimates are used by the controller to stabilize the robot motion while tracking the desired position, as described in [1].

Periodically, the operator updated the target so as to command the robot to trot a further $10\,\mathrm{cm}$ forward. The experiment continued for a total duration of $29\,\mathrm{min}$. At the end of the run, the robot had covered a total distance of about $400\,\mathrm{m}$ and trotted forward and backward 174 times. The configuration used on-line in the experiment was IMU-LO-AICP.

To measure body-relative drift we compute the average Drift per Distance Traveled (DDT) relative to the ground truth pose. The per-sample DDT is as follows:

$$\mathrm{DDT}(k) = \frac{||\Delta\bar{\mathbf{t}}_b{}^{k-N:k} - \Delta\hat{\mathbf{t}}_b{}^{k-N:k}||}{\sum_{j=k-N}^{k} ||\Delta\bar{\mathbf{t}}_b{}^{j-1:j}||} \qquad (10)$$

which is the mean absolute position drift over the period $k-N:k$ (we used $10\,\mathrm{s}$) divided by the ground truth path integral of motion of the base link (the path integral tends to overstate the distance traveled and understate DDT). For an entire run, we calculate the median of this function, which is relevant because a continuously low DDT is required for accurate footstep execution and terrain mapping. For yaw drift, we use the median absolute yaw drift per second.

In Table II, we show the results for the four configurations using the KVH 1775. One can see that the IMU-LO-VO combination reduces the DDT relative to the baseline – in particular by reducing drift in z. IMU-LO-AICP removes global drift and keeps DDT below $1\,\mathrm{cm/m}$. Using all the sensors (IMU-LO-VO-AICP combination) the drift is further reduced to $0.72\,\mathrm{cm/m}$. This result is comparable to the measurement noise of the Vicon system and satisfies our requirements.

Comparison between IMUs: We present the results for two different IMU configurations, using the industrial grade Microstrain 3DM-GX4-25 in addition to the KVH 1775. For the IMU-LO baseline, the median absolute rotation drift rate is an order of magnitude greater than for the KVH ($0.119\,°/\mathrm{s}$). However, by incorporating VO and AICP, we demonstrate that

Sensor Combination	Drift per Dist. Traveled [cm/m]					Median Yaw Drift [deg/s]
	XYZ	XY	X	Y	Z	
	KVH 1775 FOG					
IMU-LO	3.27	0.71	0.42	0.41	3.08	0.019
IMU-LO-VO	1.67	0.80	0.48	0.43	1.30	0.021
IMU-LO-AICP	0.89	0.66	0.35	0.41	0.42	0.014
IMU-LO-VO-AICP	**0.72**	**0.56**	**0.32**	**0.30**	**0.31**	**0.014**
	Microstrain GX4-25 MEMS					
IMU-LO	3.63	0.97	0.70	0.53	3.47	0.119
IMU-LO-VO-AICP	0.78	0.58	0.35	0.31	0.36	0.016

TABLE II: Median Drift per Distance Traveled (DDT) and Median Absolute Yaw Drift from Experiment 1 (see Section VII-A).

Name	**Gait**	**Duration**	**Area m^2**	**Laser**	**Ramp**
Log 1	crawl	869 s	20×5, F/B	5 RPM	✓
Log 2	crawl	675 s	20×5, F	5 RPM	✓
Log 3	trot	313 s	20×5, F/B	15 RPM	X
Log 4	trot	330 s	20×5, F/B	10 RPM	X
Log 5	trot	469 s	7×5, F/B	10 RPM	✓

TABLE III: Summary of the dataset used for Experiment 2, including log duration, size of arena, type of motion (F/B = forward/backward trajectory), laser spin rate, and terrain features.

we can reduce the rotation drift to be comparable with the KVH sensor ($0.016\,°/\mathrm{s}$). Space limitations preclude a more detailed discussion.

The results presented here show that incorporating VO reduces the drift rate relative to the base line system, while adding AICP achieves localization relative to a fixed map. So as to test performance with uneven terrain and where the reference point cloud must be updated, a second series of experiments was carried out in a larger environment.

B. Experiment 2: Comparing Variants in a Realistic Scenario

The robot explores a $20 \times 5\,\mathrm{m}^2$ industrial area (Figure 7, bottom). It navigates over uneven and rough terrain (ramps and rock beds), crawling and trotting at up to $0.5\,\mathrm{m/s}$. Turning in place (as seen in Figure 3) represents an extra challenge for the state estimation system. Lighting conditions vary dramatically during data recording, from bright light to strong shadows and from day to night-time. In some experiments, on-board lighting was used. The dataset is summarized in Table III and consists of five runs, for a total duration of $44\,\mathrm{min}$ and $300\,\mathrm{m}$ traveled.

No motion capture system is available in this space: to quantitatively evaluate the state estimation performance on the dataset, we built a prior map made up of a collection of 4 carefully aligned point clouds and we estimated drift relative to it.

Given a trajectory of estimated robot poses from an experiment, for every full laser rotation we align the point cloud to the prior map. To evaluate the accuracy of the estimated pose T_e, we can estimate the correct pose T_c from this alignment, which we assume will closely match the true ground truth pose. The error ΔT is computed as follows:

$$\Delta T = \begin{bmatrix} \Delta R & \Delta \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix} = T_e T_c^{-1} \tag{11}$$

with translation error as $||\Delta\mathbf{t}||$ and rotation error as the Geodesic distance given the rotation matrix ΔR, as in [19].

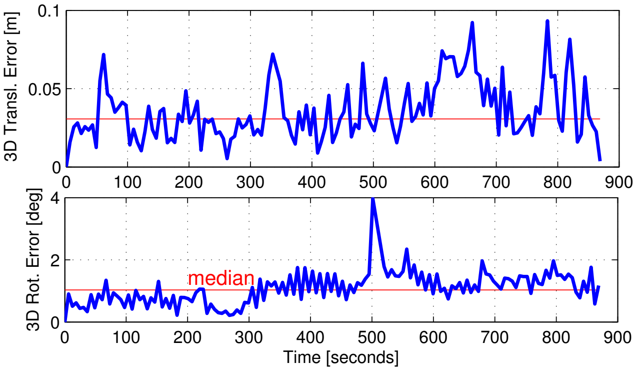


Fig. 9: Estimated error of the state estimator used in Log 1, using the configuration IMU-LO-VO-AICP. The log is referred to Experiment 2a and involved the robot crawling for a total of $40\,\mathrm{m}$.

a) Experiment 2a - Crawling Gait: In Experiment 1, we have shown (while trotting) that integrating VO reduces the pose drift rate between the lower frequency AICP corrections. Here, we focus on the importance of using VO in addition to AICP.

Figure 9 shows the estimated error over the course of Log 1, recorded in the arena of Figure 8. The robot started from pose A, reached B and returned to A. The robot crawled for $40\,\mathrm{m}$ and paused to make 3 sharp turns. The experiment was at night and used the on-board LED lights.

During this run, the reference point cloud was updated 4 times. After $860\,\mathrm{s}$, the state estimation performance had not significantly degraded, despite no specific global loop closure being computed.

In Figure 10, one can see that the median translation error was approximately $3\,\mathrm{cm}$ while the median correction made by the EKF was about $3\,\mathrm{mm}$ — both with and without VO. Because we do not observe a significant improvement in drift rates, we choose not to recommend using VO *while crawling*. This is because of the lower speed of motion and the reduced drift rate of this less dynamic gait.

b) Experiment 2b - Trotting Gait: As mentioned previously, trotting is a more dynamic gait with a higher proprioceptive drift rate, which means that the VO could better contribute when combined with AICP. Empirically, this can be seen in the inset plot in Figure 8. In this case, the algorithm with VO produces a smoother trajectory (in green) than without (in yellow). This is important because the robot's base controller uses these estimates to close position and velocity control loops. Discontinuities in the velocity estimate could lead to undesired destabilizing foot forces and controller reactions.

In brief, for the trotting logs (Logs 3, 4, 5) the integration of AICP allowed state estimation with an average 3D median translation error of approximately $4.9\,\mathrm{cm}$ (Figure 10, left). The integration of VO reduced the median translation error to $3.2\,\mathrm{cm}$. Similar behavior can be seen for the magnitude of the position correction (Figure 10, right). These results demonstrate that continuous drift has been removed and that incremental drift is minimal.

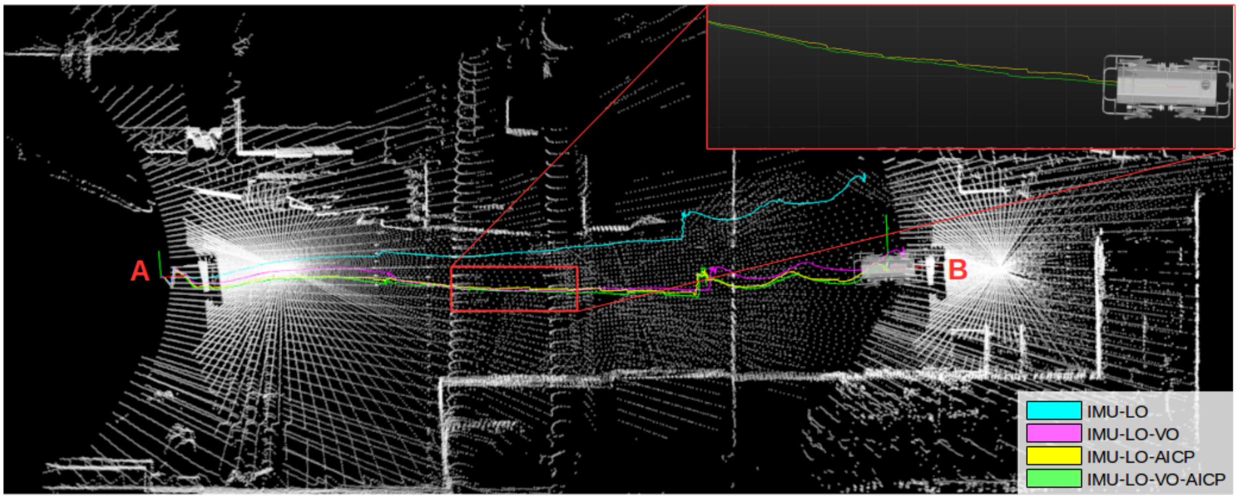


Fig. 8: A LIDAR map during a 17 m trot across the test arena. Also shown are the trajectories for the 4 estimator configurations discussed in this paper. The final combination (IMU-LO-VO-AICP) produces a smooth trajectory with continuously accurate localization (inset).

VIII. Discussion and Limitations

As described above, the proposed system is able to overcome a variety of challenges and to support accurate navigation despite the dynamic locomotion gaits. The current system limitations are: a) the incremental error introduced by updates of the reference cloud, b) the frequency of the LIDAR sensor and resulting point cloud accumulation, and c) the susceptibility of the VO system to occasionally fail during short periods of poor lighting and the absence of visual features.

The system cannot recover from a) without a SLAM or loop closure strategy. Because of the overlap analysis, AICP allows us to change reference frame rarely, meaning that the drift in the demonstrated experiments is under one centimeter.

Depending on the LIDAR spin rate, AICP corrections occur at different frequencies, while accuracy is dependent on a minimum point cloud density. Accumulating a cloud over several seconds is problematic because of the state estimator drift. At the speeds of locomotion tested here, this issue has not been limiting, however at higher speeds a higher frequency LIDAR may become necessary.

Concerning the visual odometry module, failures during experiments occurred when there was limited illumination or motion blur (*e.g.,* Figure 2). In these cases, the VO system merely resets until the next suitable frame is received.

IX. Conclusion

We have presented algorithms for the sensor fusion of inertial, kinematic, visual sensing and LIDAR to produce a reliable and consistent state estimate of a dynamically locomoting quadruped built upon a modular Extended Kalman Filter.

In particular we indicated how our approach supports dynamic maneuvers and operation in sensor impoverished situations. The reliability of our approach was demonstrated with dynamic gaits and speed up to $0.5\,\mathrm{m/s}$. A particular technical achievement has been reliably closing the loop with this state estimator in dynamic gaits. During experiments lasting over one hour, our system demonstrated to be robust and continuously accurate with drift per distance traveled below $1\,\mathrm{cm/m}$.

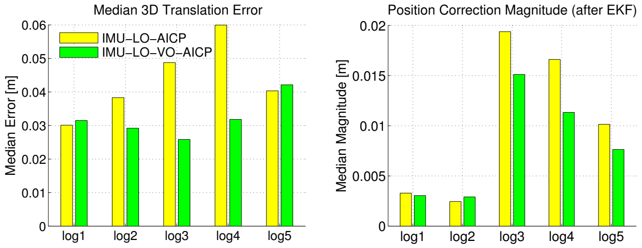


Fig. 10: Median translation error and magnitude of the AICP correction with (green) and without (yellow) visual odometry for Experiment 2 (see the dataset Table III). The smaller corrections of the IMU-LO-VO-AICP combination indicate smoother estimated trajectory.

As we move forward with our testing, we will leverage the lessons learned here in more challenging experiments. We are interested in exploring more advanced visual mapping to allow the robot to recover visual localization after events such as sharp turns. Our initial testing indicates that many visual mapping systems do not adapt well to our test scenarios.

As mentioned in Section V-B, our current filter marginalizes out previous state variables. In future work we will explore using windowed smoothing to incorporate measurements relative to previous filter states.

References

[1] V. Barasuol, J. Buchli, C. Semini, M. Frigerio, E. R. De Pieri, and D. G. Caldwell. A Reactive Controller Framework for Quadrupedal Locomotion on Challenging Terrain. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, Karlsruhe, Germany, May 2013.

[2] Timothy D. Barfoot. *State Estimation for Robotics*. Cambridge University Press, 2017.

[3] M. Blösch, M. Hutter, M. A. Höpflinger, S. Leutenegger, C. Gehring, C. D. Remy, and R. Siegwart. State

Estimation for Legged Robots - Consistent Fusion of Leg Kinematics and IMU. In *Robotics: Science and Systems (RSS)*, Sydney, Australia, July 2012.

[4] Adam Bry, Abraham Bachrach, and Nicholas Roy. State estimation for aggressive flight in GPS-denied environments using onboard sensing. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2012.

[5] M. Camurri, M. Fallon, S. Bazeille, A. Radulescu, V. Barasuol, D. G. Caldwell, and C. Semini. Probabilistic Contact Estimation and Impact Detection for State Estimation of Quadruped Robots. *IEEE Robotics and Automation Letters*, 2(2):1023–1030, April 2017.

[6] A. Chilian, H. Hirschmüller, and M. Görner. Multi-sensor data fusion for robust pose estimation of a six-legged walking robot. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, San Francisco, California, September 2011.

[7] S. Chitta, P. Vernaza, R. Geykhman, and D.D. Lee. Proprioceptive localization for a quadrupedal robot on known terrain. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, Rome, Italy, April 2007.

[8] M. F. Fallon, M. Antone, N. Roy, and S. Teller. Drift-free humanoid state estimation fusing kinematic, inertial and LIDAR sensing. In *IEEE/RSJ Int. Conf. on Humanoid Robots*, Madrid, Spain, November 2014.

[9] H. Hirschmüller. Stereo Processing by Semiglobal Matching and Mutual Information. *IEEE Trans. Pattern Anal. Machine Intell.*, 30(2):328–341, February 2008.

[10] A.S. Huang, A. Bachrach, P. Henry, M. Krainin, D. Maturana, D. Fox, and N. Roy. Visual odometry and mapping for autonomous flight using an RGB-D camera. In *Proc. of the Intl. Symp. of Robotics Research (ISRR)*, Flagstaff, USA, August 2011.

[11] M. Hutter, C. Gehring, D. Jud, A. Lauber, C. D. Bellicoso, V. Tsounis, J. Hwangbo, K. Bodie, P. Fankhauser, M. Bloesch, R. Diethelm, S. Bachmann, A. Melzer, and M. Hoepflinger. ANYmal - A Highly Mobile and Dynamic Quadrupedal Robot. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, Daejeon, Korea, October 2016.

[12] T. Koolen, S. Bertrand, G. Thomas, T. de Boer, T. Wu, J. Smith, J. Englsberger, and J. Pratt. Design of a Momentum-Based Control Framework and Application to the Humanoid Robot Atlas. *Intl. J. of Humanoid Robotics*, 13:1–34, 2016.

[13] S Lynen, M Achtelik, S Weiss, M Chli, and R Siegwart. A robust and modular multi-sensor fusion approach applied to mav navigation. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, Tokyo, Japan, 2013.

[14] J. Ma, M. Bajracharya, S. Susca, L. Matthies, and M. Malchano. Real-time pose estimation of a dynamic quadruped in GPS-denied environments for 24-hour operation. *Intl. J. of Robotics Research*, 35(6):631–653, May 2016.

[15] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós. ORB-SLAM: A versatile and accurate monocular SLAM system. *IEEE Trans. Robotics*, 31(5):1147–1163, 2015.

[16] S. Nobili, R. Scona, M. Caravagna, and M. Fallon. Overlap-based ICP tuning for robust localization of a humanoid robot. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, Singapore, May 2017.

[17] E. Olson. A passive solution to the sensor synchronization problem. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Taipei, Taiwan, October 2010.

[18] H.-W. Park, P. Wensing, and S. Kim. Online planning for autonomous running jumps over obstacles in high-speed quadrupeds. In *Robotics: Science and Systems (RSS)*, Rome, Italy, July 2015.

[19] F. Pomerleau, F. Colas, R. Siegwart, and S. Magnenat. Comparing ICP Variants on Real-World Data Sets. *Autonomous Robots*, 34(3):133–148, February 2013.

[20] C. Semini, N. G. Tsagarakis, E. Guglielmino, M. Focchi, F. Cannella, and D. G. Caldwell. Design of HyQ – A hydraulically and electrically actuated quadruped robot. *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, 225 (6):831–849, 2011.

[21] S. Shen, Y. Mulgaonkar, N. Michael, and V. Kumar. Multi-sensor fusion for robust autonomous flight in indoor and outdoor environments with a rotorcraft mav. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, Hong Kong, May 2014.