Augmentation of Mapping and Autonomous Navigation for Hexapod Robots by using a Visual Inertial System

Nitesh P Yelve, Jovan C Menezes, Shubhankar B Das and Bhavik M Panchal

Department of Mechanical Engineering, Fr. Conceicao Rodrigues Institute of Technology, Vashi, Navi Mumbai, Maharashtra-400703, India.

Abstract. In our previous work, we focused on the development of Razbot, a hexapod platform, and achieving its autonomy using the Robot Operating System (ROS). Therein, we also explained the difficulties faced with Razbot while trying to execute complex operations autonomously. Hence, in this work we address the problem and make an attempt at improving the odometry measurements of Razbot by introducing a Visual Inertial System (VIS) that includes a pair of synchronized monocular camera and inertial measurement unit which makes it capable of mapping the environment and autonomously navigating in it with increased effectiveness. To introduce the VIS, we made minor modifications in the components used and the control scheme of Razbot. In this work, we take help of the Extended Kalman Filter (EKF) for the fusion of odometry sources. Further, we also analyse the accuracy of the fused odometry by comparing the same and evaluating the errors or drifts in odometry before and after introducing the VIS in Razbot. Based on these results, we make a conclusion that the odometry source from VIS can significantly lower errors and drifts in fused odometry data, thus improving autonomous navigation and mapping of the hexapod robot. The implementation of this system will allow Razbot to be used in even more varied applications than those specified earlier.

1. Introduction
The field of research of legged robots has seen considerable advancements since the late 1990s because of their varied applications, particularly the domain of prosthetics. Bio-mimicked legged robots have shown great results in exhibiting locomotion in complex environments. While biped and quadruped robots help to achieve higher velocity of traverse, they are also accompanied with structural complexity to provide balance. Hexapod robots are relatively easier to construct as even during motion, their walking gaits ensure that at least 3 legs are in contact with the surface. Hexapods robots have been used in applications such as study of neural gait control and subsumption architectures [1], remotely operated or unmanned ground vehicles (UGVs) for rescue operations, damage inspection [2,3] and even in mimicking the motion if satellites in zero gravity [4].While initial architectural designs, such as that of the RHex [5], were aimed at achieving power autonomy, i.e. in the absence of a closed loop controller system, significant research is being carried out in the field of complete autonomy of hexapods.

First step is achieving complete autonomy is to teach the robot how to avoid obstacles and maneuver safely. Literature presents varied techniques for achieving optimum trajectory planning and navigation for hexapod robots and achieve this goal. One of the most prominent technique is the use of Fuzzy Logic. Genetic Algorithms coupled with Fuzzy Logic controllers have been used to create a Genetic Fuzzy System which provides the optimal trajectory as well as gait for motion [6,7]. Reinforcement learning
and Fuzzy Q-learning algorithms have also been used to help hexapod robots avoid obstacles and traverse [8]. Optimum trajectory can also be achieved by considering the obstacle avoidance puzzle as an optimization problem [9]. This technique however, while effective, has limitations during execution. Just as human-inspired control schemes have been used for bipedal walking robots [10], biologically inspired techniques have also been used to achieve safe locomotion for hexapod robots as well. Mechanisms based on adaptive neural control have been used to generate locomotion and simultaneously adapt the robot in changing environments and obstacles [11]. As opposed to the widespread approach of centralized control scheme, distributed control based on biological design have been incorporated in hexapod robots for to traverse uneven terrains and generate local limb reflexes [12]. Control strategies have been developed to help parallel mechanism based hexapod robots travel in dynamic or ever changing surrounding while knowing its own status and maintaining balance [13-15]. Methods used to track the location of robots usually input the positional readings from each actuator’s encoders to calculate its current state which gets iterated through a gait sequence from initialization. Such an approach is prone to error as the legs can slip or get obstructed in uneven terrains, which can lead to accumulation of errors over time. Hence, such methods are limited to static environment with smooth surfaces.

Mapping and navigation based approach using cameras is one of the many modern techniques that are used for obstacle avoidance, trajectory planning and autonomous navigation. Traditional approaches use the auto-correlation function method [16] to develop a feature detector for image regions with texture and isolated features and understand the hexapod’s environment. Data regarding the pose of the robot can be obtained by using a Scale Invariant Transform technique [17] based pose tracking algorithms which help to track the camera pose and in turn the status of the robot. Tracking cameras have also been used on hexapod robots for spatial exploration [18]. Direct Stereo Visual Odometry [19], a stereo pose tracking algorithm, applies a semi-direct monocular visual odometry method to accurately track the camera pose and the absolute scaled position of surrounding features for mapping. Such monocular camera based approach helps obtain the absolute scaled position of features easily as compared to a single camera and is effective for platform based robots, like hexapods. Generic software approaches [20] have also been developed to help robots navigate in unmodified and dynamic environments, enable short term human robot interaction, and achieve virtual telepresence. Vision based controller adaptation can be implemented on hexapod robots to autonomous navigation over uneven terrains [21,22]. It uses vision based exteroceptive terrain perception to adapt the robot’s locomotion parameters. The adaptation controller enables the robot to reactively adapt to the surface structure it is moving on.

Simultaneous Localization and Mapping (SLAM) is a versatile technique which aids in resolving many of the complications faced while implementing mapping and navigation in robots [23]. Different SLAM techniques can be implemented easily by using RGB-D cameras/sensors [24,25]. Visual–SLAM can be implemented in hexapod robots by using such RGB-D sensors. ORB–SLAM, another SLAM based versatile technique, has also been implemented on robots using a monocular or stereo or RGB-D sensor system to achieve autonomy [26,27]. Visual Planar Semantic–SLAM is another approach which proves effective while trying to achieve mapping and navigation [28]. Graph-SLAM technique can be implemented on unmanned ground vehicles using a 3D-LiDAR for mapping forests and other complex environments [29].

Out of the multiple SLAM techniques available, Visual–SLAM approach stipulates a good enough techno-economic balance. However, after implementing this approach, the quality of mapping and navigation achieved may not be optimum to perform multifaceted operations. In our previous work, we demonstrated the design and construction of our hexapod robot, Razbot [30]. We also achieved a certain degree of autonomy by incorporating Visual–SLAM coupled with the Navigation Stack of the Robot Operating System. Due to the poor quality of autonomy, in this paper, we improvise the measurement of odometry to advance the autonomy of the robot by increasing the effectiveness of mapping and navigation through the use of Visual Inertial System (VIS) with a pair of synchronized monocular camera and Inertial Measurement Unit for determining the absolute scale of robot’s pose. The use of
VIS in Razbot reduces errors or drifts in odometry, thus making autonomous navigation and mapping more robust. Commonly implemented visual inertial algorithms are Robust Visual Inertial Odometry (ROVIO) and Visual Inertial Navigation System—Monocular (VINS-Mono). While ROVIO uses a Direct Extended Kalman Filter (EKF) based approach to track multilevel patch features that are closely coupled to the underlying EKF through direct use of the intensity errors as innovation term during the update step [31], VINS-Mono, a tightly coupled and nonlinear optimization based method is used to obtain highly accurate VIO by fusing pre-integrated IMU measurements and feature observations, which are coupled with loop closure techniques that enables re-localization when the robot revisits previous landmarks [32]. As compared to ROVIO, VINS-Mono not only provides increased pose tracking accuracy and less drift but also includes an online camera IMU time offset calculation which is absent in ROVIO. Implementing VINS-Mono also increases the computer resource usage [33]. Hence, in our present research we have implemented the VINS-Mono algorithm in Razbot.

2. Design and Development

The design of the radially symmetric Razbot is explained in our earlier work. With majority of the components remaining the same, to incorporate the Visual Inertial System we have replaced the earlier used MPU 6050 IMU with the Razor 9 DoF IMU synchronized with the Arducam MT9V034 global shutter camera. To successfully implement the ROS navigation stack for autonomous operation, it is required to have the VIS with consolidated 3D depth camera. To overcome the difficulty of producing synchronized data of camera images and IMU readings, a microcontroller interfaced with IMU and a general purpose input-output pin to trigger camera at a certain interval of time. Based on the above modifications, the revised control scheme for Razbot is as seen below in figure 1.

![Figure 1: Modified control scheme for Razbot to implement the Visual Inertial System.](image)

The inverse kinematic equations of motion for Razbot, explained in our previous work along with the D-H parameters for the design, are as follows based on notation shown in table 1. The equation (1), equation (2), equation (3) and equation (4) are programmed in the Inverse Kinematics (IK) Engine to solve and provide the angle of rotation for each servo motor of Razbot.
### Table 1. List of notations.

| Sr. No. | Notation | Description |
|---------|----------|-------------|
| 1       | $x$      | X co-ordinate of the location of end-effector |
| 2       | $y$      | Y co-ordinate of the location of end-effector |
| 3       | $z$      | Z co-ordinate of the location of end-effector |
| 4       | $d_{bl}$ | Distance from body centre to coxa joint |
| 5       | $d_{cl}$ | Length of Coxa link |
| 6       | $d_{fl}$ | Length of Femur link |
| 7       | $d_{tl}$ | Length of Tibia link |
| 8       | $q_1^\circ$ | Coxa servo rotation angle |
| 9       | $q_2^\circ$ | Femur servo rotation angle |
| 10      | $q_3^\circ$ | Tibia servo rotation angle |

\[ x = \cos(q_1 + 60) \times r - \frac{d_{bl}}{2} \]  \hspace{1cm} (1)

\[ y = \sin(q_1 + 60) \times r + \frac{\sqrt{3}d_{bl}}{2} \]  \hspace{1cm} (2)

\[ z = d_{fl} \times \sin(q_2) + d_{tl} \times \sin(q_2 + q_3) \]  \hspace{1cm} (3)

\[ r = d_{cl} + d_{tl} \times \cos(q_2 + q_3) + d_{fl} \times \cos(q_2) \]  \hspace{1cm} (4)

### 3. Software Architecture

#### 3.1. Visual Inertial System

The VIO algorithm and Visual–SLAM require hardware synchronization of the camera and IMU at the millisecond level. To achieve this, we have used the Arduino for synchronization of the camera and IMU. Arduino calculates precise millisecond timestamps for each IMU measurement at a frequency of 200 Hz. Arduino then triggers the camera via the trigger line to capture a new image at 20 Hz frequency. Thus, after every 10 measurements of the IMU, one camera frame is captured. Timestamp and sequence number data are sent to the onboard processing unit’s synchronization node. This synchronization node receives IMU data from the Arduino and publishes images by matching the sequence number and then stamps time on the image frame according to the received time stamp in a ROS camera topic. This process is illustrated in figure 2.

#### 3.2. Signal Processing

The next step is the processing of sensor data and fusing it by using the EKF. Odometry data is obtained from two sources, one from the gait system that calculates the odometry of the robot by integrating the strides of the legs which is done iteratively throughout right from initialization. Another source of odometry is from the VIS running VINS-Mono as explained in section 3.1. The odometry from the gait system is calculated at a rate of 100 Hz and is termed as raw odometry. On the other hand, the VIO from VINS-Mono has an update rate of 10 Hz. Odometry from both these sources is fused using the EKF. This is achieved by integrating the ROS package Robot Localization and is shown in figure 3. Robot localization is a collection of state estimation nodes, each of which is an implementation of a nonlinear state estimator for robots moving in 3D space [34]. The EKF localization node of the robot localization package is an implementation of the EKF. It uses an omnidirectional motion model to project the state...
forward in time and corrects that projected estimate using perceived sensor data. The sensor data is referred to as odometry sources which are used by the EKF to correct the predicted state. After fusion of the two odometry sources using the EKF, Razbot is tested for a random path. Figure 4 shows fused odometry calculated by the EKF (green line) and features registered in the test environment (white dots).

**Figure 2:** Camera-IMU synchronization flow chart for the VINS-Mono.

**Figure 3:** Fusion of Odometry data in the Robot Localization package using the EK Filter.

**Figure 4:** Odometry filtered data.
3.3. Mapping and Navigation
As explained in our previous work, the concept of Visual-SLAM (or simply V-SLAM) is used for 3D mapping. This SLAM technique builds the surrounding’s map and at the same time enables the robot to localize itself with a focus on real-time operation. V-SLAM fuses all depth data from the sensor into a volumetric dense model that is used to track the camera pose using Iterative Closest Point (ICP) method. The RTAB-Map algorithm of ROS, used to map the environment, relies on the robot’s transformation data set (obtained from the filtered odometry and calculated using robot localization EKF) as well as the depth data from Kinect. It then progressively generates a map as it explores the environment and localizes itself based on key features. This technique of loop closure removes any anomaly and drifts in maps, thus, obtaining a fine 3D map of the environment.

After generating a map of the environment, the robot can now autonomously navigate through the mapped environment while the localization loop is still active. For navigation, to summarize the details specified in our previous work, Razbot calculates the cost map from the generated map and then proceeds to plan its path. Path planning is done by the move-base-flex node which links together a global and local planner to accomplish its global navigation task [35]. The navigation phase is carried out by implementing the ROS Navigation Stack. The point cloud data developed is visualized in ROS as shown in figure 5:

![Figure 5: 3D point cloud map generated by Razbot.](image)

4. Experimental Analysis and Results
The experiment is carried out to evaluate the performance and accuracy of Razbot achieved by incorporating VIS in it. The experiment is conducted in an indoor environment by following a complex random trajectory. Figure 6 depicts the actual start and end points of the trajectory completed by the robot in a grid cell of size 500 x 500 mm each. Several random circular trajectories are used to examine the drift by comparing it with the actual trajectory. Razbot follows the depicted trajectory based on velocity commands. There are three main points that are taken into consideration for determining the accuracy of VIO. The ‘Actual Start/End’ point, which denotes the true start and end point of the closed loop trajectory followed by Razbot; ‘Odom End’ point that denotes the end point of the fused odometry; and ‘Odom Raw End’ which denotes the end point of the raw odometry generated from the gait system. We have analyzed two odometry sources in the experiment i.e. raw odometry which is calculated from the gait system and fused odometry that is calculated from the EKF fusion of the VIO algorithm and raw odometry. In figure 6, the red path indicates fused odometry and the blue path indicates raw odometry. This odometry data is further analyzed in a graphical form for evaluation of accuracy. The X and Y co-ordinates of the trajectory, obtained using the two odometry sources, are plotted against time as shown in figure 07 and figure 08, respectively. Actual trajectory starts at (0,0) and ends at (0,0). The
values of the end points from the odometry sources are compared with the actual end points of the trajectory for estimation of accuracy. Table 2 indicates the error estimation for the values obtained from the two odometry sources and the percentage reduction in error in the two directions after fusion of the VIO algorithm using the EKF.

**Figure 6:** Graphical depiction of circular trajectory (Red path: Fused odometry, Blue path: Raw odometry).

**Figure 7:** X-co-ordinate of Razbot vs. time (red path: Fused odometry, blue path: Raw odometry).

**Figure 8:** Y-co-ordinate of Razbot vs. time (red path: Fused odometry, blue path: Raw odometry).
Table 2: Error estimation for circular trajectory.

| Direction | Fused Odometry Error (m) | Raw Odometry Error (m) | Difference in Error (m) | Percentage reduction in Error (%) |
|-----------|--------------------------|------------------------|-------------------------|----------------------------------|
| X         | 0.1                      | 0.3                    | 0.2                     | 66.67                            |
| Y         | -0.25                    | -0.3                   | 0.05                    | 16.67                            |

From the above table, it is derivative that there is a decrease in the error due to drift by an amount of 66.67% in the X-direction and comparatively by a smaller amount of 16.7% in the Y-direction. Similar experiment was carried out for a rectangular trajectory (shown in figure 9) with a much higher reduction in error of 91.1% in the X-direction and 44.4% in the Y-direction. Table 3 depicts the estimation of error for the rectangular trajectory.

![Graphical depiction of rectangular trajectory](image)

Table 3: Error estimation for rectangular trajectory.

| Direction | Fused Odometry Error (m) | Raw Odometry Error (m) | Difference in Error (m) | Percentage reduction in Error (%) |
|-----------|--------------------------|------------------------|-------------------------|----------------------------------|
| X         | 0.04                     | 0.45                   | 0.41                    | 91.1                             |
| Y         | -0.15                    | -0.36                  | 0.16                    | 44.4                             |

Upon analysis of the graphical trajectories and evaluation of the error in drift, it can be inferred that the overall performance of autonomous navigation of Razbot is aggrandized by the implementation of the VINS-Mono algorithm. In addition, as the complexity of the trajectory reduces, the drift error is also proportionately reduced leading to an augmentation in Razbot’s ability to perform autonomously.

5. Conclusion
In this work we implemented the VINS-Mono algorithm on our hexapod robot, Razbot, to demonstrate the effect of fusion of the EKF-based Visual Inertial Odometry Source on the accuracy of the odometry and autonomous behavior of the robot. While we developed Razbot in our previous work, we noticed the inability of the robot to perform navigation autonomously quickly and smoothly. To solve this issue,
we incorporated the VIS consisting of a camera and IMU that provides the odometry source which is fused with the raw odometry data generated by the gait system. To confirm the accuracy of the newly incorporated system, we conducted tests by moving Razbot in two contours (circular and rectangular).

As per our expectations, we observed reduction in the drift error in all test cases; even as high as 91.1% (in the X-direction for rectangular trajectory). From this we can conclude that the 3D map developed by the robot has much less drift after addition of the VIS as more accurate and filtered odometry data is obtained for use of mapping and autonomy in an environment. Additionally, the hexapod platform could localize itself and navigate autonomously in the environment more efficiently and with reduced lag time. This enables the robot to accomplish more exacting tasks such as autonomous exploration missions in challenging environments and traversing in locations hazardous for humans. The authors believe that the extension of such techniques may prove to be useful to enable legged robots navigate hazardous terrain and autonomously sense their location relative to the surface and modify their trajectory as needed.

A drawback that one may face while utilizing this approach is that the VIS may lead to erratic estimations of odometry values during conditions such as bad illumination and motion blur. An easy solution to this is the use of sophisticated or industrial grade IMU for sensitive real world autonomous missions. Authors may address these above stated issues in future work. From the above test results, for solution to this is the use of sophisticated or industrial grade IMU for sensitive real world autonomous estimations of odometry values during conditions such as bad illumination and motion blur.

References

[1] Mills J W 1992 Stiquito: A small, Simple, Inexpensive Hexapod Robot Indiana University Computer Dept. Technical Report Rev. A 363 pp 1–48
[2] Karma S et al. 2015 Use of unmanned vehicles in search and rescue operations in forest fires: Advantages and limitations observed in a field trial Int. J. of Disaster Risk Reduction 13 pp 307–12
[3] Smith T B, Barreiro J, Smith D E, Sunspiral V and Chavez-Clemente D 2008 ATHLETE’s feet: Multi-resolution planning for a hexapod robot Proc. of ICAPS-08 Scheduling and Planning Apps. Workshop (Sydney Australia)
[4] NASA tumblr page, https://nasa.tumblr.com/post/175779680559/6-fun-facts-about-our-new-hexapod-robot last accessed on 2021/05/09
[5] Saranli U, Buehler M and Koditschek D E 2000 Design, modeling and preliminary control of a compliant hexapod robot Proceedings of IEEE Int. Conf. on Robotics and Automation (San Francisco California United States of America) pp 2589–96
[6] Roy S S and Pratihar D K 2014 Kinematics, dynamics and power consumption analyses for turning motion of a six legged robot J. of Intelligent and Robotic Systems: Theory and Applications 74(3–4) pp 663–88
[7] Pratihar D K, Deb K and Ghosh A 2002 Optimal path and gait generations simultaneously of a six legged robot using a GA-fuzzy approach Robotics and Autonomous Systems 41(1) pp 1–20
[8] Hong J, Tang K and Chen C 2018 Obstacle avoidance of hexapod robots using fuzzy Q-learning 2017 IEEE Symp. Series on Computational Intelligence, SSCI 2017 – Proc. pp 1–6
[9] Zhao Y, Chai X, Gao F and Qi C 2018 Obstacle avoidance and motion planning scheme for a hexapod robot Octopus-III Robotics and Autonomous Systems 103 pp 199–212
[10] Ames A 2014 Human-Inspired Control of Bipedal Walking Robots IEEE Trans. on Automatic Control 59 pp 1115–30
[11] Goldschmidt D, Wörgötter F and Manoonpong P 2014 Biologically-inspired adaptive obstacle negotiation behavior of hexapod robots Frontiers in Neurorobotics pp 1–16
[12] Espenschied K S, Quinn R D, Beer R D and Chiel H J 1996 Biologically based distributed control
and local reflexes improve rough terrain locomotion in a hexapod robot. Robotics and Autonomous Systems 18(1–2) pp 59–64

[13] Xin G, Deng H, Zhong G and Wang H 2015 Dynamic analysis of a hexapod robot with parallel leg mechanisms for high payloads Proc. of 10th Asian Control Conf., ASCC pp 1–6

[14] Zhong G, Deng H, Xin G and Wang H 2016 Dynamic hybrid control of a hexapod walking robot: experimental verification IEEE Trans. Ind. Electron 63 pp 5001–11

[15] Deng H, Xin G, Zhong G and Mistry M 2017 GaIt and trajectory rolling planning and control of hexapod robots for disaster rescue applications Robotics and Autonomous Systems 95 pp 13–24

[16] Harris C and Stephens M A 1998 Combined corner and edge detector Proc. of the Alvey Vision Conf., Alvey Vision Club pp 147–51

[17] Lowe D 2004 Distinctive image features from scale-invariant keypoints Int. J. of Computer Vision 60(2) pp 91–110

[18] Bayer J and Faigl J 2019 On autonomous spatial exploration with small hexapod walking robot using tracking camera intel RealSense T265 2019 European Conf. on Mobile Robots, ECMR 2019 – Proc. pp 1–6

[19] Holzmann T, Fraundorfer F and Bischof H 2016 Direct stereo visual odometry based on lines Proc. of Joint Conf. on Computer Vision, Imaging, and Computer Graphics Theory and Application (Rome Italy) pp 474–85

[20] Thrun S et al. 1999 MINERVA: A second-generation museum tour-guide robot Proc. of IEEE Int. Conf. on Robotics and Automation (Detroit Michigan United States of America)

[21] Bjelonic M, Homberger T, Kottege N, Borges P, Chli M and Beckerle P 2017 Autonomous navigation of hexapod robots with vision-based controller adaptation Proc. - IEEE Int. Conf. on Robotics and Automation pp 5561–68

[22] Bjelonic M, Kottege N, Homberger T, Borges P, Beckerle P and Chli M 2018 Weaver: Hexapod robot for autonomous navigation on unstructured terrain J. of Field Robotics 35(7) pp 1063–79

[23] Leonard J J and Durrant-Whyte H F 2002 Simultaneous map building and localization for an autonomous mobile robot Proc. of IEEE/RSJ Int. Workshop on Intelligent Robots and Systems (Osaka Japan) vol 3 pp 1442–47

[24] Cizek P and Faigl J 2016 On localization and mapping with RGB-D sensor and hexapod walking robot in rough terrains 2016 IEEE Int. Conf. on Systems, Man, and Cybernetics, SMC 2016 – Conf. Proc. pp 2273–78

[25] Omara H I M A and Sahari K S M 2015 Indoor mapping using kinect and ROS 2015 Int. Symp. on Agents, Multi-Agent Systems and Robotics, ISAMSR 2015 pp 110–16

[26] Mur-Artal R, Montiel J M M and Tardos J D 2015 ORB-SLAM: A versatile and accurate monocular SLAM system IEEE Trans. on Robotics 31(5) pp 1147–63

[27] Mur-Artal R and Tardos J D 2017 ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras IEEE Trans. on Robotics

[28] Bavle H, De La Puente P, How J P and Campoy P 2020 VPS-SLAM: Visual planar semantic SLAM for aerial robotic systems IEEE Access 8 pp 60704–18

[29] Pierzchala M, Gigüère P and Astrup R 2018 Mapping forests using an unmanned ground vehicle with 3D LiDAR and graph-SLAM Computers and Electronics in Agriculture, 145 (February 2017) pp 217–25

[30] Menezes J, Das S, Panchal B, Yelve N, Kumar P 2020 Mapping, Trajectory Planning, and Navigation for Hexapod Robots using ROS Proc. of 1st Int. Conf. on Energy, Materials Sciences & Mechanical Engineering., EMSME-2020 (Delhi India)

[31] Bloesch M, Omari S, Hutter M and Siegwart R 2015 Robust visual inertial odometry using a direct EKF-based approach Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (Hamburg Germany)

[32] Qin T, Li P and Shen S 2018 VINS-Mono: A robust and versatile monocular visual-inertial state
estimator IEEE Trans. on Robotics 34(4) pp 1004–20

[33] Delmerico J and Scaramuzza D 2018 A benchmark comparison of monocular visual-inertial odometry algorithms for flying robots Proc. of IEEE Int. Conf. on Robotics and Automation (Brisbane Australia) pp 2502–09

[34] Moore T and Stouch D 2016 A generalized extended Kalman filter implementation for the robot operating system Advances in Intelligent Systems and Computing 302

[35] Putz S, Simon J and Hertzberg J 2018 Move base flex Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (Madrid Spain) pp 3416–21