Comprehensive Simulation of Quadrotor UAVs Using ROS and Gazebo

Johannes Meyer[1], Alexander Sendobry[1], Stefan Kohlbrecher[2], Uwe Klingauf[1], and Oskar von Stryk[2]

[1] Department of Mechanical Engineering, TU Darmstadt, Germany
[2] Department of Computer Science, TU Darmstadt, Germany

Abstract. Quadrotor UAVs have successfully been used both in research and for commercial applications in recent years and there has been significant progress in the design of robust control software and hardware. Nevertheless, testing of prototype UAV systems still means risk of damage due to failures. Motivated by this, a system for the comprehensive simulation of quadrotor UAVs is presented in this paper. Unlike existing solutions, the presented system is integrated with ROS and the Gazebo simulator. This comprehensive approach allows simultaneous simulation of diverse aspects such as flight dynamics, onboard sensors like IMUs, external imaging sensors and complex environments. The dynamics model of the quadrotor has been parameterized using wind tunnel tests and validated by a comparison of simulated and real flight data. The applicability for simulation of complex UAV systems is demonstrated using LIDAR-based and visual SLAM approaches available as open source software.

1 Introduction

Quadrotor UAVs have successfully been used both in research and for commercial applications in recent years. Impressive results have been shown using quadrotor aircraft of various sizes and in different scenarios. The inherently instable nature of quadrotor flight can lead to loss or damage of UAVs easily, especially when evaluating prototype soft- or hardware. The lack of a simulation environment for quadrotor UAVs that covers realistic flight dynamics, camera and range sensors and an easy integration with existing robotic middleware solutions motivated this work. We present a comprehensive framework to simulate our quadrotor, that has been developed during the last few years. It is based on the Gazebo open source simulator and the Robot Operating System (ROS), that has become a de facto standard in robotics research and facilitates integration of contributions by other researchers. Common sensors for autonomous robots like LIDAR devices, RGB-D and stereo cameras are already available for Gazebo and can be attached to the robot, while plugins for other, more UAV-specific sensors like barometers, GPS receivers and sonar rangers have been added as part of this work.

The remainder of this paper is organized as follows: After the discussion of related work in section 2, section 3 presents the simulation model considering

I. Noda et al. (Eds.): SIMPAR 2012, LNAI 7628, pp. 400–411, 2012.

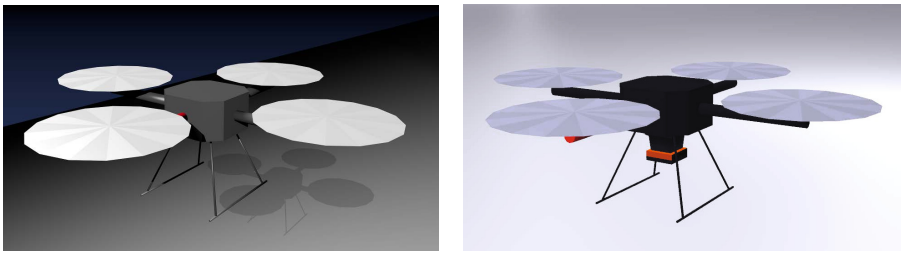
geometry, flight dynamics and control and how the model is implemented in Gazebo. Comparative results from flight tests and simulation runs as well as a demonstration of applicability for evaluating high-level algorithms are presented in section 4.

2 Related Work

As we aim at a comprehensive approach for simulation of quadrotor UAV systems, we provide an overview both of simulation/ground truth tracking approaches as well as quadrotor control approaches. Most approaches for quadrotor simulation focus on vehicle dynamics for controller design, often using specialized tools like Matlab/Simulink [20]. Sometimes other tools like the Flightgear open source simulation framework are used for visualization [18,9]. Using such approaches, testing of sensor-based high level control and behaviors is not possible or requires significant additional implementation effort. Quadrotor UAVs can be simulated using USARsim [7], but a recently published ROS integration [2] is of limited scope. In [1], the use of a simulator also providing sensor data is mentioned, but not made available for testing.

Several authors have proposed dynamics models for the simulation of quadrotor aircraft which are based on the same flight mechanical principles [10,3,9,11,20]. While dealing with different aspects in detail, none of them considers motor and propeller dynamics, aerodynamics, external disturbances (e.g. wind), and noisy sensor signals and state estimation in an integrated fashion.

Recently, external optical tracking for the acquisition of ground truth data has been used with great success [8,17]. The installation of such systems however is costly and often not feasible due to space constraints. Even if such a system is available, testing of multi-UAV control approaches in simulation is advantageous, as potential collisions or other faults incur no cost in simulation.



(a) (b)

Fig. 1. Mesh-based quadrotor model: (a): Model shown rendered in Blender. (b) Model used in Gazebo. A Hokuyo UTM-30LX laser scanner is mounted below the main body.

3 Model Description and Simulation

As we aim at comprehensive simulation of all relevant components including low level sensing, system dynamics and control, we provide an overview of these parts independently. Gazebo provides a multi-robot simulation environment including dynamics simulation, which is provided by the ODE or bullet physics engines. While the simulator considers gravity, contact forces and friction by its own, it does not cover aerodynamics and propulsion systems that are especially required for aerial vehicles. A plugin systems enables the user to add custom controllers for simulated robots and sensors or to influence the environment.

3.1 Geometry

The robot geometry has been modeled using the open source software Blender. To be able to provide different colors (both texture or material based) for the model, the visual geometry is provided using the COLLADA format, while the collision geometry is modeled as a .stl mesh. The model is designed to have a low polygon count and still retain the relevant aspects of quadrotor geometry. As a trade-off between visual fidelity, collision detection and dynamics modeling needs, the propellers are modeled as discs.

3.2 Dynamics Model

One of the main advantages of the quadrotor concept is the simplicity of its propulsion and steering system, consisting only of four independent motors and propellers with fixed pitch, where each pair of opposite propellers rotates in one direction to avoid yaw torque during roll and pitch movements. As a result, the overall system dynamics are mainly determined by the thrust and torque induced by the individual motor/propeller units.

Flight Dynamics. The movement of a rigid body can be described by the sum of all forces $\boldsymbol{F}$ and torques $\boldsymbol{M}$ acting on the vehicle:

$$\dot{\boldsymbol{p}}^{\mathrm{n}} = \boldsymbol{v}^{\mathrm{n}} \tag{1a}$$

$$\dot{\boldsymbol{v}}^{\mathrm{n}} = m^{-1}\mathbf{C}_{\mathrm{b}}^{\mathrm{n}}\boldsymbol{F} \tag{1b}$$

$$\dot{\boldsymbol{\omega}}^{\mathrm{b}} = \boldsymbol{J}^{-1}\boldsymbol{M} \tag{1c}$$

Here, $\boldsymbol{p}^{\mathrm{n}}$ and $\boldsymbol{v}^{\mathrm{n}}$ are the position and velocity of the body's center of gravity in the (inertial) navigation coordinate system, $\boldsymbol{\omega}^{\mathrm{b}}$ is its angular rate given in body coordinates and $\mathbf{C}_{\mathrm{b}}^{\mathrm{n}}$ is the rotation matrix that transforms a vector from body (index b) to navigation coordinates (index n).

The mass m and inertia $\boldsymbol{J}$ of the quadrotor need to be known and have been estimated by weighing the individual components and using the geometric model. The force vector $\boldsymbol{F}$ comprises motor thrust $\boldsymbol{F}_{\mathrm{M}}$, drag forces $\boldsymbol{F}_{\mathrm{d}}$ and the gravity vector $\boldsymbol{F}_{\mathrm{g}}$. The torque vector $\boldsymbol{M}$ is divided into propulsion torque $\boldsymbol{M}_{\mathrm{M}}$ and drag moments $\boldsymbol{M}_{\mathrm{d}}$. Drag forces and moments are given by:

$$\boldsymbol{F}_{\mathrm{d}} = -\boldsymbol{C}_{\mathrm{d,F}} \cdot \mathbf{C}_{\mathrm{n}}^{\mathrm{b}} \cdot |\boldsymbol{v}^{\mathrm{n}} - \boldsymbol{v}_{\mathrm{w}}^{\mathrm{n}}| \, (\boldsymbol{v} - \boldsymbol{v}_{\mathrm{w}}) \tag{2a}$$

$$\boldsymbol{M}_{\mathrm{d}} = -\boldsymbol{C}_{\mathrm{d,M}} \cdot \left|\boldsymbol{\omega}^{\mathrm{b}}\right| \boldsymbol{\omega}^{\mathrm{b}} \tag{2b}$$

with the diagonal drag coefficient matrices $\boldsymbol{C}_{\mathrm{d,F}}$ and $\boldsymbol{C}_{\mathrm{d,M}}$ and the wind vector $\boldsymbol{v}_{\mathrm{w}}^{\mathrm{n}}$. Finally, the gravity force is given by

$$\boldsymbol{F}_{\mathrm{g}} = m \cdot \mathbf{C}_{\mathrm{n}}^{\mathrm{b}} \cdot \begin{bmatrix} 0 & 0 & g_e \end{bmatrix}^{\mathrm{T}} \quad . \tag{3}$$

With these forces and torques resulting from self motion of any system in space and the propulsion forces and torques described in the following section, the vehicle movement can be obtained by solving equations (1).

Motor Dynamics. The propulsion system of our quadrotor UAV consists of four brushless DC motors. The dynamic behavior of a brushless DC motor has been derived from [12] with some simplifications. Assuming a very low inductance of the motor coils, the current rise time can be neglected. The motor dynamic behavior therefore simplifies to a PT_1 element and is described by Eqs. (4) - (6). In steady state the induced anchor voltage U_{A} depends on the rotation speed $\boldsymbol{\omega}_{\mathrm{M}}$ and the anchor current I_{A}:

$$U_{\mathrm{A}} = R_{\mathrm{A}} I_{\mathrm{A}} + \Psi \boldsymbol{\omega}_{\mathrm{M}} \tag{4}$$

The electromagnetic torque M_{e} for each motor is given by

$$M_{\mathrm{e}} = \Psi I_{\mathrm{A}} \tag{5}$$

With the mechanical torque M_{m} and the motor inertia J_{M} the change in rotation speed can be calculated through:

$$\dot{\omega}_{\mathrm{M}} = \frac{1}{J_{\mathrm{M}}} \cdot (M_{\mathrm{e}} - M_{\mathrm{m}}) = \frac{1}{J_{\mathrm{M}}} \cdot \left(\frac{\Psi}{R_{\mathrm{A}}} \cdot (U_{\mathrm{A}} - \Psi \boldsymbol{\omega}_{\mathrm{M}}) - M_{\mathrm{m}} \right) \tag{6}$$

The nonlinear term M_{m} describes the torque resulting from bearing friction as well as load friction (i.e. drag) of the airscrew. It can be written as $M_{\mathrm{m}} = k_{\mathrm{T}} \cdot T$ where T is the thrust of a single airscrew [15] which is a a broad simplification of a former approach [21] without loss of accuracy.

Thrust Calculation. In contrast to the former approach we now use a nonlinear quadratic approximation for thrust calculation, similar to [9]. This approach has been selected based on wind tunnel tests (cf. Fig. 2(a)) and is sufficiently accurate as to not require the use of more complex thrust models [10]. With the dynamic expression of the motor's rotational speed $\boldsymbol{\omega}_{\mathrm{M}}$ from equation (6) it is straightforward to calculate the thrust force T for a single motor-airscrew combination:

$$T = C_{\mathrm{T},0} \boldsymbol{\omega}_{\mathrm{M}}^2 + C_{\mathrm{T},1} v_1 \boldsymbol{\omega}_{\mathrm{M}} \pm C_{\mathrm{T},2} v_1^2 \tag{7}$$

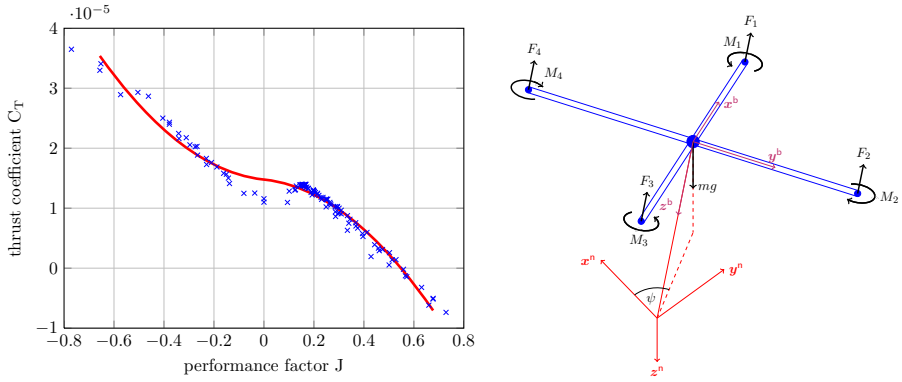


Fig. 2. (a) Thrust coefficient C_T of an airscrew as a function of its performance factor J. Crosses mark wind tunnel measurements, while the solid line represents the approximation. (b) Sketch of the quadrotor to show the different coordinate systems and naming conventions.

Dividing the above equation by $\boldsymbol{\omega}_\mathrm{M}^2$ and using the performance factor $J = v_1/\boldsymbol{\omega}_\mathrm{M}$ the thrust coefficient $C_\mathrm{T}(J)$ is given by:

$$C_\mathrm{T}(J) = C_{\mathrm{T},0} + C_{\mathrm{T},1} J \pm C_{\mathrm{T},2} J^2 \tag{8}$$

where the parameters $C_{\mathrm{T},i}$ have been identified in wind tunnel test. A negative v_1 (meaning a falling quadrotor) results in a positive prefix of $C_{\mathrm{T},2}$. In Fig. 2(a) the polynomial approximation of $C_\mathrm{T}(J)$ is shown. For a quadrotor helicopter the free stream velocity v_1 in general is different for each of the rotors. It can be calculated through geometric inspection of the vehicle shown in Fig. 2(b):

$$(v_1)_i = -\begin{bmatrix}0 & 0 & 1\end{bmatrix} \cdot \left(\boldsymbol{v}^\mathrm{b} + \left(\boldsymbol{\omega}^\mathrm{b} \times \boldsymbol{e}_i\right) \cdot l_\mathrm{M}\right) \tag{9}$$

with the unit vectors

$$\boldsymbol{e}_1 = \begin{bmatrix}1 & 0 & 0\end{bmatrix}^\mathrm{T}, \qquad \boldsymbol{e}_2 = \begin{bmatrix}0 & 1 & 0\end{bmatrix}^\mathrm{T}, \qquad \boldsymbol{e}_3 = -\begin{bmatrix}1 & 0 & 0\end{bmatrix}^\mathrm{T}, \quad \boldsymbol{e}_4 = -\begin{bmatrix}0 & 1 & 0\end{bmatrix}^\mathrm{T}$$

for the four different motors. l_M is the distance between the geometric centers of motors and quadrotor. With the coordinate system conventions shown in Fig. 2(b) the following expression for the overall wrench of the quadrotor can be determined:

$$\boldsymbol{F}_\mathrm{M}^\mathrm{b} = \begin{bmatrix}0\\0\\-\Sigma_{i=1}^4 F_i\end{bmatrix} \qquad \boldsymbol{M}_\mathrm{M}^\mathrm{b} = \begin{bmatrix}(F_4 - F_2)\cdot l_\mathrm{M}\\(F_1 - F_3)\cdot l_\mathrm{M}\\-M_1 + M_2 - M_3 + M_4\end{bmatrix} \tag{10}$$

The four single forces F_i are calculated by solving equation (7) while the moments M_i are obtained through combining equation (4) and (5). The incorporation of

blade flapping effects which can be used to aid state estimation [4] is subject of future work.

We implemented two plugins that calculate propulsion and drag forces acting on the aircraft given the internal state of the vehicle, the four motor voltages and the wind vector. The current wind can be specified as constant vector or provided by an external model or from real log data. Gazebo then applies the calculated forces and torques to the quadrotor body for each simulation step.

3.3 Sensor Simulation

As attitude, position and velocity cannot be measured directly, accurate models are needed to simulate the signals from various sensors needed for estimating the state of the UAV. These sensors have been implemented as independent Gazebo plugins and can be attached to the model by including them in the robot URDF description. The plugins accept parameters covering the error characteristics and the WGS84 position, altitude and orientation of the Gazebo reference frame in the world coordinate system wherever necessary.

Error Model. All sensors share a common first order Gauss Markov error model [5], permitting simulation of sensors with different error characteristics. Each simulated measurement $y(t)$ at time t is given by

$$y = \hat{y} + b + w_y \tag{11a}$$

$$\dot{b} = -\frac{1}{\tau} b + w_b \tag{11b}$$

where $\hat{y}$ is the true value or vector, b is the current bias and w_y and w_b are independent, zero-mean white Gaussian noise variables. w_y is additive noise acting directly on the measurement and w_b describes the characteristics of the random drift with time constant τ.

Inertial Measurement Unit. The inertial measurement unit (IMU) is the most important sensor for the stabilization of quadrotor flight as it measures the angular velocities and accelerations of the vehicle body in the inertial frame. Integration of these values provides a good reference of attitude and speed over short time intervals with fast response times, but is not suitable for long-term reference due to the significant drift of available low-cost sensors. Also note that an observer onboard the vehicle cannot distinguish gravity from other external forces and therefore the acceleration of the body in the world frame cannot be measured directly without knowing the orientation of the body.

Barometric Sensor. For simulating the static pressure at the present altitude, we use the International Standard Atmosphere (ISA) model as defined by the International Civil Aviation Organization (ICAO), which describes the pressure, temperature and density of the earth's atmosphere under average conditions at mid latitudes. The elevation of the simulation reference frame above mean sea level and the simulated pressure (only required for the output of pressure values in hPa) at sea level can be specified as parameters.

Ultrasonic Sensor. For controlling the height during the takeoff and landing phases and for switching on and off the motors, the range estimate from an downward pointing ultrasonic sensor is used. This device transmits short ultrasound impulses and returns the distance corresponding to the first echo returned from the ground or an object within it's field of view. Available ultrasound sensors have a maximum range of about 3 to 6 meters. The simulated ultrasonic sensor uses the Gazebo ray sensor interface to determine ray-casting based distances to world geometry. The distance value returned is the minimum of all rays (9 by default).

Magnetic Field Sensor. The earth magnetic field serves as a reference for the heading or yaw angle of the quadrotor. As using a single axis compass would lead to significant errors with increasing roll and pitch angles, three-axis magnetometers are commonly used for UAVs. With the assumption that the earth-fixed magnetic field vector is constant within the area of operation, it is straightforward to calculate the body-fixed vector given the declination, inclination and field magnitude. Deviation errors through interference from parts of the robot itself are covered by the generic error model.

GPS Receiver. Pseudo range measurements and the resulting position and velocity solution are influenced by different factors like the satellite ephemeris errors, atmospheric errors or receiver errors [19]. These error sources are approximated using the Gauss-Markov error model, with the parameters of our uBlox receiver module having been determined experimentally. To reproduce the interdependency of position and velocity errors we use the noise-affected velocity measurement error $(\boldsymbol{v}_{\mathrm{GPS}} - \hat{\boldsymbol{v}}_{\mathrm{GPS}})$ instead of $\boldsymbol{w_b}$ in Eq. (11b) for the integration of the position error. A more detailed consideration of GPS errors and especially multipath effects in the vicinity of buildings is left for future work. To calculate WGS84 coordinates from the simulated position and velocity in Cartesian coordinates we use a simple equirectangular projection that is based on a flat world assumption. This projection is accurate enough in the vicinity of the chosen reference point and outside the polar regions.

3.4 State Estimation and Control

Although state estimation and control are not specific to simulation, both components are required to close the loop between simulated sensor signals and the resulting motor voltages required to stabilize and control the quadrotor.

For estimating the state of the system we use an Extended Kalman Filter (EKF) to fuse all available measurements to a single navigation solution containing the orientation, position and velocity of the vehicle as well as observable error states like the IMU bias errors. This approach is usually referred to as integrated navigation.

Our controller is implemented as a set of cascaded PID controllers, with the inner loop controlling the attitude, yaw rate and vertical velocity and an outer loop controlling the horizontal velocity, heading and altitude (Fig. 3). This

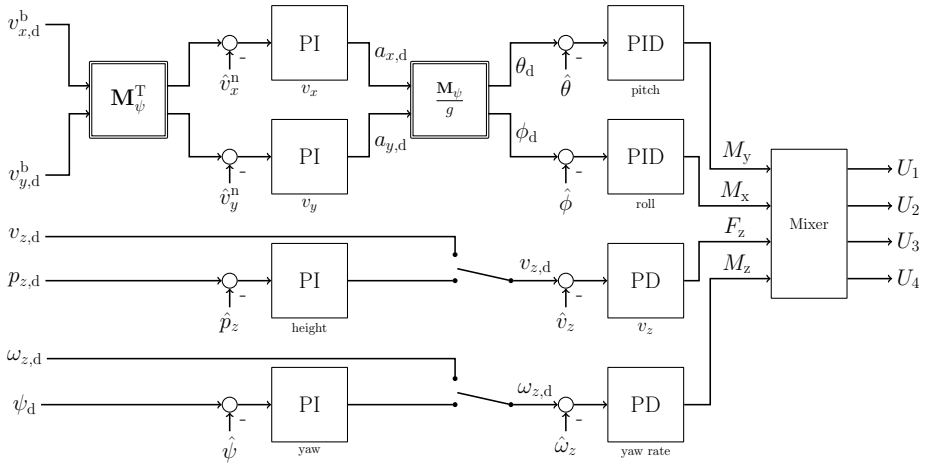


Fig. 3. The controller is realized through separate cascaded PID controllers controlling roll and pitch movement, yaw rate and vertical velocity

approach assumes that each axis and the altitude can be controlled independently, which is valid for moderate deviations from the hovering state. The output of the inner loop are commanded torques and vertical thrust, which are translated to motor voltages either by using a static mixture matrix or by feeding them into an inverted model of the propulsion system presented in section subsection 3.2.

For simulation we use exactly the same implementation as on the real quadrotor. It is based on the Open Robot Control Software (Orocos) toolchain [6], which provides interfaces to ROS and executes tasks satisfying hard realtime constraints on the onboard PC system. This software-in-the-loop approach offers great flexibility for testing advanced control algorithms before the deployment on the real vehicle and therefore minimizes the risk of damage or loss dramatically. Implementation details can be found in previous publications [16].

4 Experiments

Different aspects of simulation are validated using experiments in this section. We also show examples of comprehensive simulation scenarios using the flight dynamics model as well as leveraging existing ROS open source software.

4.1 Validation of Dynamics Model

To validate the dynamics model, we let both the real and simulated UAV perform a test trajectory consisting of transitions between different velocities. All measurable variables of the real quadrotor show the same characteristics as the corresponding simulated counterparts. The power spectrum densities (PSD) of

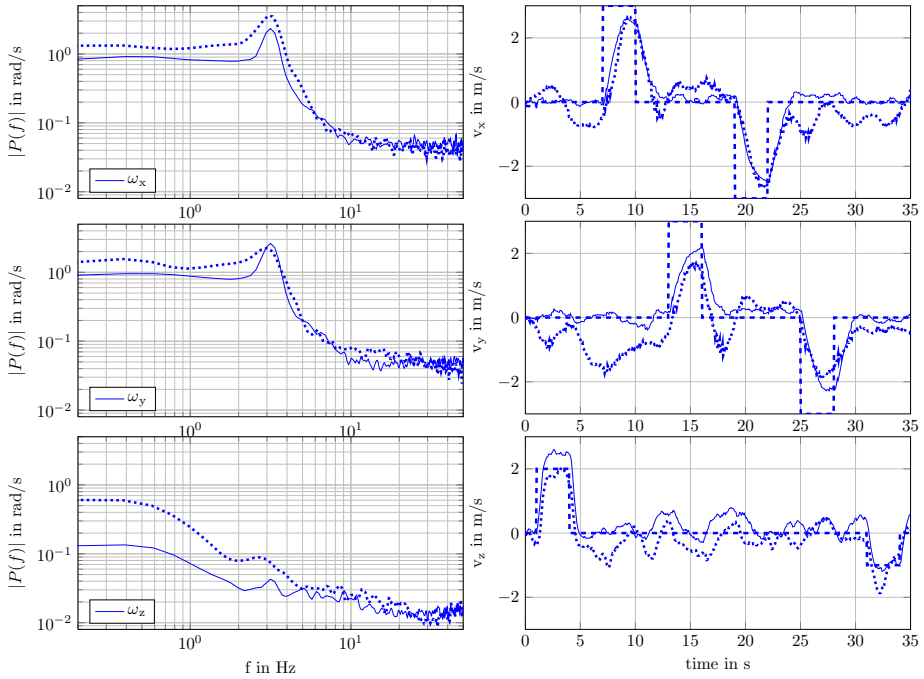
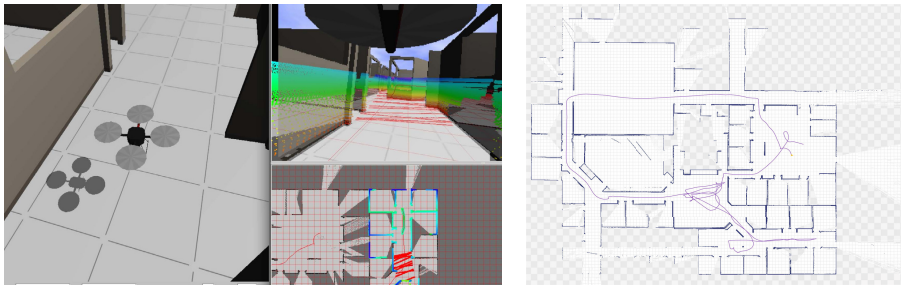
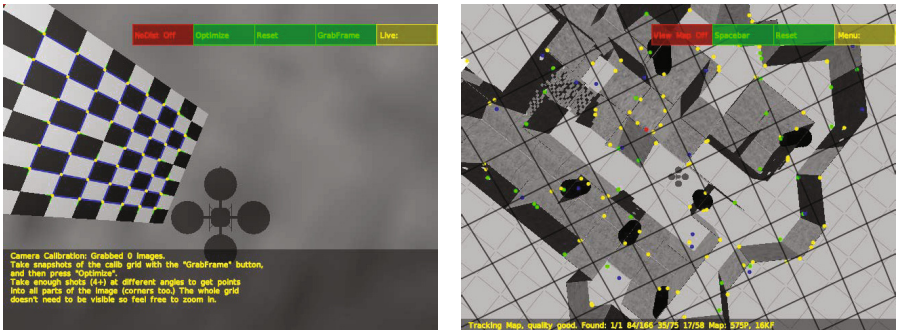


Fig. 4. Diagrams of simulated and measured angular and translational velocity. Dotted lines represent measurements while solid lines are simulated data. The left side shows the PSD of the angular rates. On the right side the estimated velocity both in simulation and reality with the commanded speeds (dashed line) is shown.



(a) (b)

Fig. 5. Indoor SLAM simulation: (a): Screenshot of the GUI. On the left the Gazebo simulation environment is visible. On the top right the view of the forward facing camera is shown, with LIDAR point cloud and map data projected into the image. A top down ortho view is visible on the bottom right (b) Final map generated after teleoperation of the UAV through the scenario.



(a) (b)

Fig. 6. Visual SLAM simulation: (a): Calibration of camera system in simulation. (b) Screenshot of PTAM being used for visual SLAM on a quadrotor hovering above a simulated NIST standard arena for response robots.

the angular rates and and the velocities are shown in Fig. 4. The controller and a dead time of about 15 ms cause the quadrotor to oscillate slightly with a frequency of about 3 Hz which is easily visible in the frequency domain. Differences in velocity between simulation and reality are mainly due to a gusty wind of about 5 m/s which was apparent during the outdoor tests. In simulation, we therefore defined a constant wind of 5 m/s.

4.2 Example Scenarios

In this section, different example scenarios are shown, demonstrating the comprehensive nature of quadrotor simulation and the interfacing with other open source ROS software. Instructions for reproducing all presented scenarios are provided on the *hector_quadrotor*[1] website on ros.org.

In- and Outdoor Flight Scenarios. We flew the simulated quadrotor through two example indoor and outdoor worlds to evaluate the quality of high-level sensor data. Using the estimated state or ground truth data, the quadrotor pose can be visualized along with sensor data.

To demonstrate the applicability for indoor SLAM simulation, we deploy a previously developed SLAM approach [14] on the quadrotor UAV. The Willow Garage office environment is part of the Gazebo ROS package, demonstrating the applicability and interoperability of the quadrotor simulation with existing Gazebo environments. The quadrotor UAV is teleoperated using a gamepad for this demonstration. As shown in Fig. 5(a), sophisticated visualization including projection of visualization data into camera images is possible by leveraging available ROS tools like *rviz*. The final map learned is shown in Fig. 5(b) and of comparable quality to those learned in real world scenarios.

A video of outdoor flying is available online[2].

[1] http://www.ros.org/wiki/hector_quadrotor
[2] http://www.youtube.com/watch?v=9CGIcc0jeuI

Visual SLAM. To demonstrate simulated image based state estimation, we deploy a modified version [22] of the original PTAM system [13] for visual SLAM. As demonstrated in Fig. 6(a), checkerboard-based calibration of camera parameters can also be performed in simulation. Fig. 6(b) shows a screenshot of the PTAM GUI while the simulated quadrotor UAV hovers above an example scenario, successfully tracking features in the image and estimating the aircraft pose. It should be noted that the default camera simulation in Gazebo is of limited fidelity as it does not exhibit effects like motion blur.

5 Conclusion

We presented a framework for the simulation of quadrotor UAV systems employing ROS and the Gazebo simulator. The tight integration with existing (and future) ROS tools permits the comprehensive simulation of quadrotor UAVs including low level sensing, flight dynamics and external sensing using any sensor available for Gazebo simulation. The level of detail can be adapted depending on the application, e.g. by using ground truth data for control or bypassing the propulsion model.

Acknowledgments. This work has been funded by the Research Training Group 1362 "Cooperative, Adaptive and Responsive Monitoring in Mixed-Mode Environments" of the German Research Foundation (DFG).

References

1. Achtelik, M., Bachrach, A., He, R., Prentice, S., Roy, N.: Autonomous navigation and exploration of a quadrotor helicopter in GPS-denied indoor environments. In: Robotics: Science and Systems Conference (2008)
2. Balakirsky, S.B., Kootbally, Z.: USARSim/ROS: A Combined Framework for Robotic Control and Simulation. In: ASME 2012 International Symposium on Flexible Automation (ISFA 2012). ASME (2012)
3. Bouabdallah, S., Siegwart, R.: Full control of a quadrotor. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 153–158 (November 2007)
4. Bristeau, P., Callou, F., Vissière, D., Petit, N., et al.: The navigation and control technology inside the AR Drone micro UAV. In: 18th IFAC World Congress, Milano, Italy, pp. 1477–1484 (2011)
5. Brown, R., Hwang, P., et al.: Introduction to random signals and applied Kalman filtering. Wiley, New York (1992)
6. Bruyninckx, H.: Open robot control software: the OROCOS project. In: IEEE International Conference on Robotics and Automation (ICRA), vol. 3, pp. 2523–2528. IEEE (2001)
7. Carpin, S., Lewis, M., Wang, J., Balakirsky, S., Scrapper, C.: USARSim: a robot simulator for research and education. In: IEEE International Conference on Robotics and Automation (ICRA), pp. 1400–1405 (2007)

8. Ducard, G., D'Andrea, R.: Autonomous quadrotor flight using a vision system and accommodating frames misalignment. In: IEEE International Symposium on Industrial Embedded Systems (SIES), pp. 261–264. IEEE (2009)
9. Goel, R., Shah, S., Gupta, N., Ananthkrishnan, N.: Modeling, Simulation and Flight Testing of an Autonomous Quadrotor. In: IISc Centenary International Conference and Exhibition on Aerospace Engineering, ICEAE, Bangalore, India, pp. 18–22 (2009)
10. Hoffmann, G.M., Huang, H., Wasl, S.L., Tomlin, E.C.J.: Quadrotor helicopter flight dynamics and control: Theory and experiment. In: AIAA Guidance, Navigation, and Control Conference (2007)
11. Huang, H., Hoffmann, G., Waslander, S., Tomlin, C.: Aerodynamics and control of autonomous quadrotor helicopters in aggressive maneuvering. In: IEEE International Conference on Robotics and Automation (ICRA), pp. 3277–3282 (May 2009)
12. Isermann, R.: Mechatronische Systeme: Grundlagen (German Edition). 1. Auflage 1999, 1. korrigierter Nachdruck - Studienausgabe edn. Springer (December 1999)
13. Klein, G., Murray, D.: Parallel tracking and mapping for small AR workspaces. In: 6th IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR), pp. 225–234. IEEE (2007)
14. Kohlbrecher, S., Meyer, J., von Stryk, O., Klingauf, U.: A Flexible and Scalable SLAM System with Full 3D Motion Estimation. In: IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR). IEEE, Kyoto (2011)
15. Leishman, G.: Principles of Helicopter Aerodynamics, 2nd edn. Cambridge Aerospace Series. Cambridge University Press (April 2006)
16. Meyer, J., Strobel, A.: A flexible real-time control system for autonomous vehicles. In: 41st International Symposium on Robotics (ISR) and 6th German Conference on Robotics (ROBOTIK). VDE (2010)
17. Michael, N., Mellinger, D., Lindsey, Q., Kumar, V.: The GRASP Multiple Micro-UAV Testbed. IEEE Robotics Automation Magazine 17(3), 56–65 (2010)
18. Qiang, Y., Bin, X., Yao, Z., Yanping, Y., Haotao, L., Wei, Z.: Visual simulation system for quadrotor unmanned aerial vehicles. In: 30th Chinese Control Conference, pp. 454–459 (July 2011)
19. Rankin, J.: An error model for sensor simulation GPS and differential GPS. In: Position Location and Navigation Symposium, pp. 260–266. IEEE (1994)
20. Rodić, A., Mester, G.: The Modeling and Simulation of an Autonomous Quad-Rotor Microcopter in a Virtual Outdoor Scenario. Acta Polytechnica Hungarica 8(4) (2011)
21. Sendobry, A.: A Model Based Navigation Architecture for Small Unmanned Aerial Vehicles. In: European Navigation Conference. Royal Institute of Navigation (RIN) (November 2011)
22. Weiss, S., Scaramuzza, D., Siegwart, R.: Monocular-SLAM–based navigation for autonomous micro helicopters in GPS-denied environments. Journal of Field Robotics 28(6), 854–874 (2011)