

SimTwo as a simulation environment for flight robot dynamics evaluation

Guilherme Amaral^{1,3}, Paulo Costa^{2,3}

¹Department of Electrical Engineering, Faculty of Engineering University of Porto, Porto, Portugal (up201406865@fe.up.pt); ²Faculty of Engineering University of Porto, Portugal (paco@fe.up.pt); ³INESC TEC Institute for Systems and Computer Engineering of Porto, Portugal

Abstract

Aerial Robots has become very popular in robotics groups. This ongoing interest is driven by a significant number of potential end-user applications where it is necessary to reduce human intervention. The validation of control/cooperation algorithms not always is an easy task and requires a large number of humans to keep tests safety. Is present in this paper the simulation of a quadrotor model in SimTwo simulation environment and evaluated the flight dynamic of the robot. To do that was developed a control layer responsible to the hovering maneuver and “go to” position function. The results prove that the objectives were achieved successfully and the simulation was validated. The navigation simulation is left as future work.

Subject Headings. Robotics, simulation model, digital control.

Author Keywords. Simulation, Unmanned Aerial Vehicle, control.

1. Introduction

The human interview complexity in the natural environment has originated a growing need for performance risk tasks, i.e., inspection actions of infra-structures either environment itself. Robots are many times called to replace human interview in actions that requires precision or are dangerous.

Some complex tasks require the necessity to obtain wide visual information (or other sensory information) through multiple observations points. In this context, mobile autonomous aerial robots arise as an important and valuable tool.

Thinking about a scenario with a team of aerial robots, for a safety reasons, it is necessary to have at least one human operator for each robot, which is not feasible for some research groups (Echeverria et al. 2012; Folgado et al. 2007; Dewan et al. 2013b; Lächele et al. 2012; Michael and Kumar 2008; Mira 2007; Costa et al. 2011). A reasonable solution is the utilization of Simulation Software's. An example of this is the work from Johannes (Meyer et al. 2012) with a simulation environment developed in Gazebo and integrated with Robotic Operating System (ROS) that was very useful to evaluate the aerial robot behavior such as flight dynamic. Also developed in Gazebo was the Nathan (Koenig et al. 2004) work that simulates a distributed formation of a swarm team of ground robots with the ability to adapt the shape of the formation based on the environment constrains. The work from Dewan (Dewan et al. 2013a) to evaluate the proposed optimization method to perform cooperative exploration between heterogeneous vehicles was also implemented in a simulation environment.

In this article is present the simulation of a quadrotor using the realistic simulator SimTwo (Costa et al. 2011). The real vehicle, modeled in simulator, is illustrated in the Figure 1.



Figure 1: Asctec Pelican Micro Aerial Vehicle

The hovering maneuver was achieved successfully as well as the implementation of a “go to” point controller.

The outline of the paper follows. In section 2 is approached the State of the Art as regard quadrotor simulators. Section 3 presents some assumptions used in the system modeling. Section 4 is related to evolved referentials in this application. Section 5 is dedicated to control architecture presentation. The implementation in SimTwo simulator is addressed in section 6. For final, the conclusions are presented in section 7.

2. State of the Art

Due the difficult to evaluate algorithms in aerial robots without damage itself, the use of simulators is a path rather sought by researches. External environment conditions as wind and dynamics effects are features that giving realism to the simulator. A simulator should be as realistic as possible to minimize the difference between the simulation and the practice.

There are many commercials and open-source simulators. Even though 2D simulators, such as Enki (Magenat 2013), Stage (Hedges 2005) and BugWorks (Thornton 2006), we will focus in 3D simulators. All of them, listed below, already offer quadrotor models that can be used for simulations.

2.1. Gazebo

Gazebo is a simulator very popular under Robotic Community and has evolved a lot since it was integrated in Robotic Operating System (ROS) platform (Koenig et al. 2004). The lack of documentation is a negative point for this software that requires a big effort to sensors integration into the vehicles and to develop realistic outdoor scenarios. However, the Gazebo 4.0 version (released in July 2014), provides a graphical user interface that could overcome the limitations mentioned.

2.2. USARSim

Initially developed as a simulator for search and rescue operations, Unified System for Automation and Robot Simulation is widely used in RoboCup competitions (Carpin et al. 2007). It follows a modular component approach and uses the Unreal Engine gaming platform. The disadvantage of this software is the external communication layer that doesn't support some of the most common robotics middleware without additional programming.

2.3. Webots

Webots is a commercial simulator that even though provides a full programming environment to create robots and environments, has an unintuitive and complex interface to construct new robots and components (Michel et al. 2004).

2.4. MORSE

The Modular Open Robots Simulation Engine is a powerful simulator that runs on the Blender Game Engine to interface 3D environment with external robotics software. Developed in Python, it has a lot of sensors that are fully supported by the middleware and can provide data as their real world counterparts allowing also the choice of different levels of realism and abstraction. MORSE proved to be very versatile, modular, flexible and reusable (Echeverria et al. 2012).

2.5. SimTwo

SimTwo is a realistic simulator system that can support several types of robots (Costa et al. 2011). With this complete tool is possible to design robots and environments as well as define objects characteristics imperials to a realistic behavior. Devoted to an educational and research approach this simulator being a versatile robot simulation environment that allows rapid test and design of differential, omnidirectional, industrial, humanoid robots, etc., developed in Object Pascal. The dynamics realism in SimTwo is obtained by decomposing a robot in parts such as rigid bodies and electric motors (Costa et al. 2011).

3. System Modeling

A quadrotor is described as a vehicle with 4 thrusters in a cross configuration. The two pairs of thrusters (1,3) and (2,4), as shown in the Figure 2, spins in opposite directions. Through the speed variation, there is a variation in lifting force thereby producing movement. In this way, increasing/decreasing the speed of four thrusters simultaneously, create vertical movement.

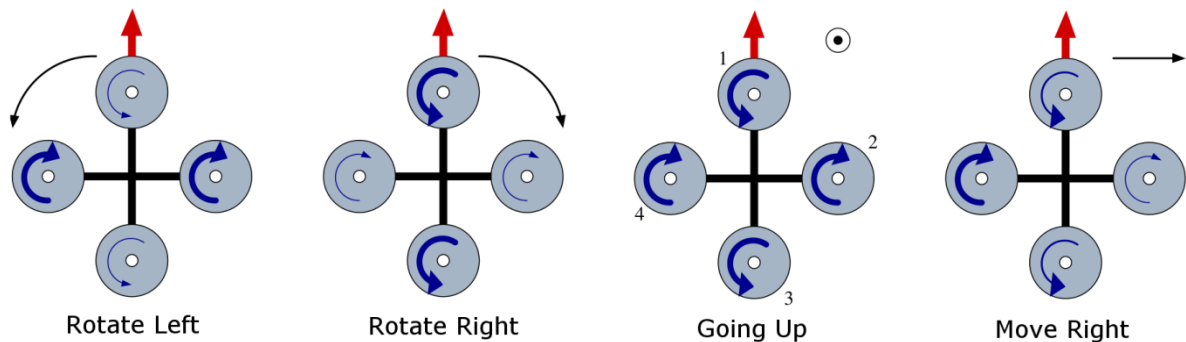


Figure 2: Quadrotor movements' description – Arrows dimensions are proportional to propeller spin speed

To actuate in pitch moment is necessary to decrease/increase the speed rotation of motor 1 and increase/decrease the speed rotation of motor 3 from the same number of magnitude. In the same way, to actuate on roll moment, is necessary to decrease/increase the speed rotation of motor 2 and increase/decrease the speed rotation of motor 4.

The yaw moment is achieved ranging the speed rotation difference between thrusters pairs 1/3 and 2/4.

4. Referentials

The referential concept identifies a coordinator system, as always necessary the utilization of, at least, two referentials: one for vehicle representation and another for navigation referential representation.

The utilization of multiple sensors that provides data in different referentials, create the need of coordinates transformation for one unique referential in order to do the sensory fusion. Below are presented the used referentials.

4.1. ECEF (Earth-Centered, Earth-Fixed)

The ECEF referential (Xu et al. 2007), used by GPS, is a Cartesian coordinates system (x,y,z) presented in Figure 3. Some characteristics of this referential are:

- The referential origin is coincident with the Earth's center of mass;
- The Z axis is coincident with the Earth rotation axis;
- The X axis is coincident with the interception of planes defined by the Greenwich Meridian and Ecuador;
- The plane defined by X and Y axis coincides with the Ecuador plane.

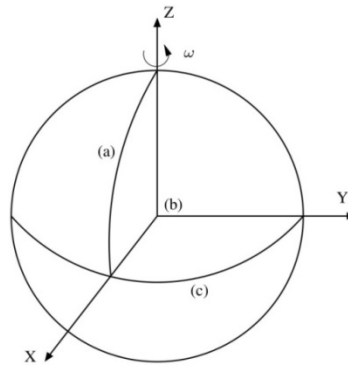


Figure 3: ECEF Referential - (a) Greenwich Meridian, (b) Earth's center of mass, (c) Ecuador line

4.2. Vehicle Referential

The body referential, represented in Figure 4, has its axis coincident with the vehicle referential. The xx' axis points to the front of aircraft (roll axis), the yy' points to the aircraft left (pitch axis) and the zz' points to up (yaw axis), completing an orthogonal coordinates system. The origin of this referential is considered in the cross of coincident symmetry, in this case, with the vehicle's center of mass.

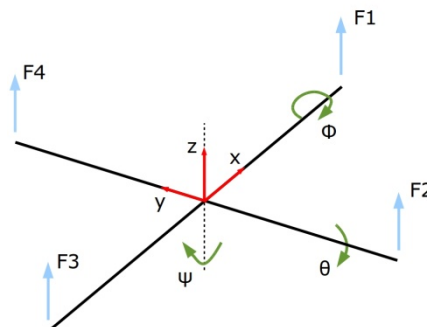


Figure 4: Vehicle referential

4.3. Relationship between vehicle referential and ECEF referential

Through Figure 4 we can observe the vehicle referential. We should assume that the body center of mass is coincident with the origin of the referential. In order to list both referentials is necessary the introduction of a rotation matrix R. This matrix is constituted through successive rotations of three Euler angles (Grewal et al. 2007) known as roll, pitch

and yaw (ϕ , θ , ψ , respectively). In this way obtain E through successive rotations multiplied by system coordinates B that is mathematical described as:

$$E = R \cdot B \quad (1)$$

Considers up for the Earth referential $E = \{X,Y,Z\}$ and the body referential $B = \{x,y,z\}$. For a real-time control, at each moment is needed the position knowledge and orientation of B related to E.

The rotation of a rigid body could be parameterized using some methods (Euler angle, quaternion, etc.). The Tait-Bryan angles, also known as “Cardan angles”, are used in aeronautic engineering. Those angles correspond to a specific set of Euler angles, that is, consists in a three Euler angles (Grewal et al. 2007) with a pre-defined rotation convention. In this article will be used the same nomenclature used in the aeronautic engineering. The Euler angles, represented in Figure 5 will be designated in the following way:

$$\begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix} = \begin{bmatrix} \text{roll} \\ \text{pitch} \\ \text{yaw} \end{bmatrix} \quad (2)$$

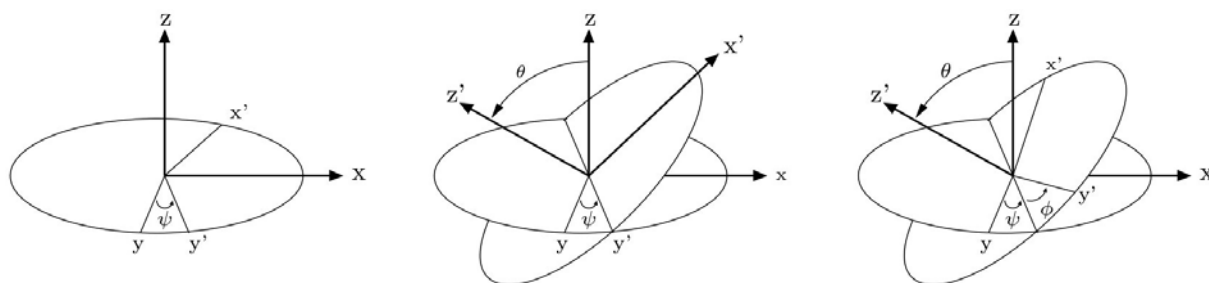


Figure 5: Tait-Bryan angles

According to the axis convention adopted, the rotation matrices will be the following, respectively to roll, pitch and yaw (Perez et al. 2007; Bouabdallah et al. 2007):

$$R_x(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi & \cos \phi \end{bmatrix} \quad (3)$$

$$R_y(\theta) = \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix} \quad (4)$$

$$R_z(\psi) = \begin{bmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (5)$$

The complete rotation matrix, as well known as Direct Cosine Matrix, is:

$$R(\phi, \theta, \psi) = \begin{bmatrix} \cos \psi \cdot \cos \theta & \cos \psi \cdot \sin \theta \cdot \sin \phi - \sin \psi \cdot \cos \theta & \cos \psi \cdot \sin \theta \cdot \cos \phi + \sin \psi \cdot \sin \phi \\ \sin \psi \cdot \cos \theta & \sin \psi \cdot \sin \theta \cdot \sin \phi + \cos \psi \cdot \cos \theta & \sin \psi \cdot \sin \theta \cdot \cos \phi - \cos \psi \cdot \sin \phi \\ -\sin \theta & \cos \theta \cdot \sin \phi & \cos \theta \cdot \cos \phi \end{bmatrix} \quad (6)$$

5. Control

In the control and basic stabilization of the vehicle will be used a decoupled approach assuming that the system is little disturbed relatively his balanced point from what can be conjugate effects from four separately controllers: height, pitch, yaw and roll.

In this way, the stabilization problem consists in the project of 4 controllers each of one responsible to stabilize one variable (height, roll, pitch and yaw). This controllers generates a reference (U_1, \dots, U_4) correspondent a generalized control force that after is combined in order to obtain the reference speeds for each motor (Amaral et al. 2009). The control architecture is represented in the figure below.

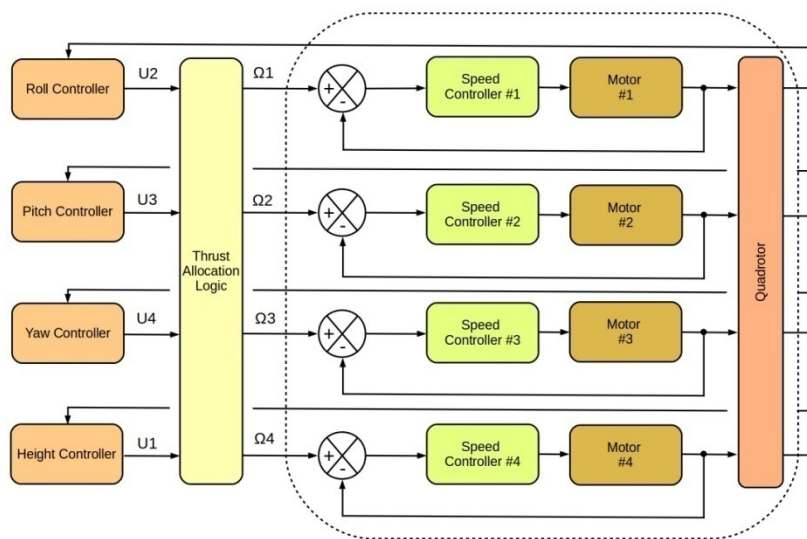


Figure 6: Quadrotor control architecture

Each speed controller was hand tuned to obtain a satisfactory step response.

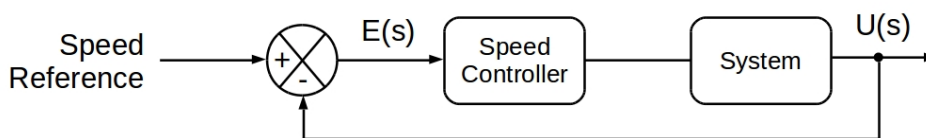


Figure 7: Block diagram of a generic speed controller

The difference between the desired and actual speed is injected in the speed controller that try to generate a reference in order to minimize or, in the best case, cancel the error (Figure 7).

6. Simulator

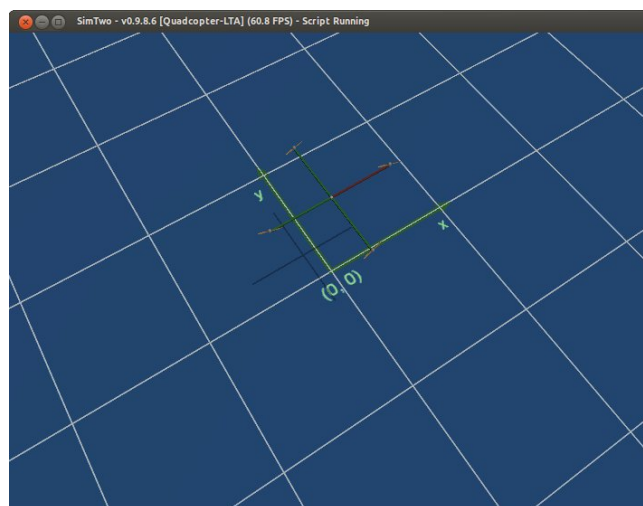


Figure 8: SimTwo World View Window

This simulator can combine a high level of reality with a reduced computational power and with good graphical representation instead another simulators.

The main robot is constituted by a central sphere and four arms arranged in such a manner that confines a cross disposal. In the exterior arms' extremity are present motors with propellers joined through "Hinge" joints. Objects parameters as well as weights, drag coefficients, thrust coefficients, position, etc., are defined in a XML dedicated file.

The controller code was developed in Pascal.

A particularity of this simulator is that at program beginning is called an initializing function where some variables are loaded and then control loop start to run at 25 Hz (by default).

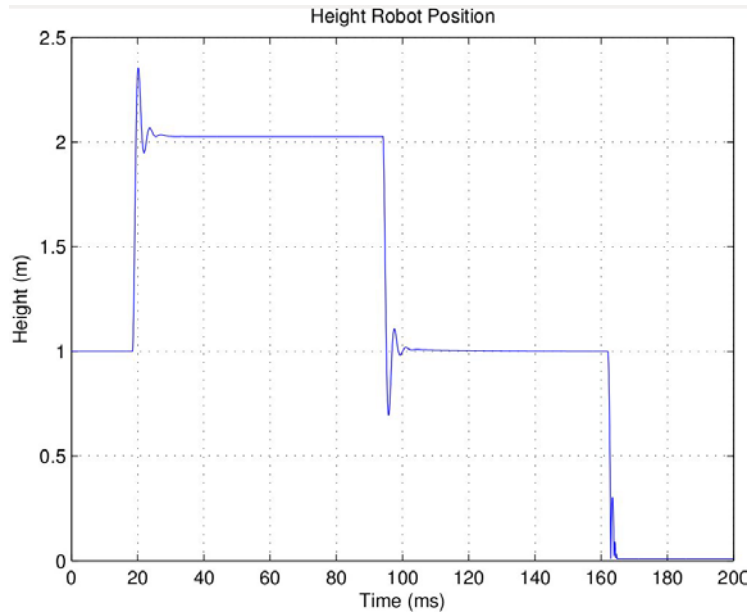


Figure 9: Evolution of vehicle's height for different set points

For the height controller evaluation was defined different values for the desired height, namely the sequence {1, 2, 1, 0}. The behavior of the vehicle's height is shown in the Figure 9.

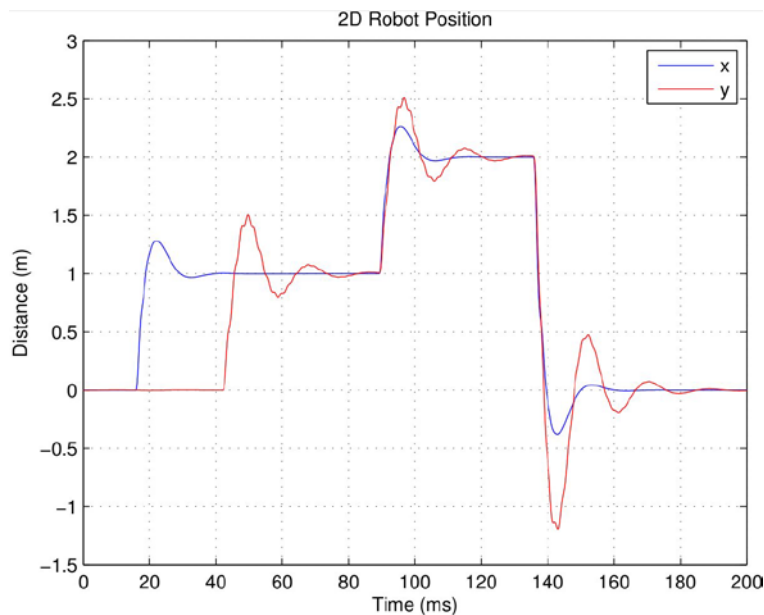


Figure 10: Evolution of vehicle's position for different set points

Figure 10 illustrates the position controller behavior. Taking advantage of the “go to” developed function, were given different points in the Earth referential. The function calculates control actions using actual and desired position in order vehicle's position converge to given points. We can observe an overshoot near references changes that can be perfectly accepted taking into account the vehicle dynamic. From the origin, was defined the set point (1,0) and then (1,1). After the vehicle has reached the position, was defined, at same time, a variation in both axes: (2,2). Finally, was defined the origin position (0,0). All tests were done with height reference of 1 meter.

7. Conclusions

In this paper we propose the simulation of a quadrotor vehicle like using the SimTwo Simulator software. We present the relationship between different two referentials and use it to implement the control layer, responsible to keep the robot in the air. A “go to” point function was also implemented and validated successfully. The vehicle dynamics influence is notorious in the quadrotor behavior denouncing the power of the simulator and the right values assigned to objects parameters.

For future work is the application of a claw in the quadrotor structure confining the capacity to transport objects from one side to another and simulation of navigation layer.

References

- Amaral, Guilherme. 2009. “Desenvolvimento E Controlo de Um Veículo Autónomo Aéreo Quadrotor.” Master Thesis. Instituto Superior de Engenharia do Porto.
- Bouabdallah, Samir. 2007. “Design and Control of Quadrotors with Application to Autonomous Flying.” PhD. Thesis. École polytechnique fédérale de Lausanne EPFL. DOI: [10.5075/epfl-thesis-3727](https://doi.org/10.5075/epfl-thesis-3727).
- Carpin, Stefano, Mike Lewis, Jijun Wang, Stephen Balakirsky, and Chris Scrapper. 2007. “USARSim: A Robot Simulator for Research and Education.” In *IEEE International Conference on Robotics and Automation*, 10-14.
- Costa, Paulo, José Gonçalves, Jose Lima, and Paulo Malheiros. 2011. “SimTwo Realistic Simulator: A Tool for the Development and Validation of Robot Software.” *International Journal of Theory and Applications of Mathematics & Computer Science* 1: 17-33. <http://uav.ro/applications/se/journal/index.php/TAMCS/article/view/3>.
- Dewan, Ayush, Aravindh Mahendran, Nikhil Soni, and K. Madhava Krishna. 2013a. “Heterogeneous UGV-MAV Exploration Using Integer Programming.” *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. Ieee, 5742-49. DOI: [10.1109/IROS.2013.6697188](https://doi.org/10.1109/IROS.2013.6697188).
- . 2013b. “Heterogeneous UGV-MAV Exploration Using Integer Programming.” In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 5742-49. IEEE. DOI: [10.1109/IROS.2013.6697188](https://doi.org/10.1109/IROS.2013.6697188).
- Echeverria, Gilberto, Séverin Lemaignan, Arnaud Degroote, Simon Lacroix, Michael Karg, Pierrick Koch, Charles Lesire, and Serge Stinckwich. 2012. “Simulating Complex Robotic Scenarios with MORSE.” *SIMPAR'12 Proceedings of the Third International Conference on Simulation, Modeling, and Programming for Autonomous Robots*, May. Ieee, 197-208. DOI: [10.1109/ICRA.2011.5980252](https://doi.org/10.1109/ICRA.2011.5980252).
- Folgado, Encarnacion, Mariano Rincon Zamorano, José Ramón Álvarez Sánchez, and José Mira. 2007. “A Multi-Robot Surveillance System Simulated in Gazebo.” In *International Work-Conference on the Interplay Between Natural and Artificial Computation*, 202-11. DOI: [10.1007/978-3-540-73055-2_22](https://doi.org/10.1007/978-3-540-73055-2_22).
- Grewal, Mohinder S., Lawrence R. Weill, and Angus P. Andrews. 2007. “*Global Positioning Systems, Inertial Navigation, and Integration*”. New Jersey: John Wiley and Sons. ISBN: 978-0-470-09971-1.
- Hedges, Reed et al. 2005. "Stage: 2D multiple robot simulator". Accessed February 18th, 2015. <http://playerstage.sourceforge.net/stage/stage.html>.

- Koenig, Nathan, and Andrew Howard. 2004. "Design and Use Paradigms for Gazebo, an Open-Source Multi-Robot Simulator." *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems* 3. Ieee: 2149-54. DOI: [10.1109/IROS.2004.1389727](https://doi.org/10.1109/IROS.2004.1389727).
- Lächele, Johannes, Antonio Franchi, Heinrich H. Bulthoff, and Paolo Robuffo Giordano. 2012. "SwarmSimX: Real-Time Simulation Environment for Multi-Robot Systems." In *3rd International Conference on Simulation, Modeling, and Programming for Autonomous Robots*, 375-87. DOI: [10.1007/978-3-642-34327-8_34](https://doi.org/10.1007/978-3-642-34327-8_34).
- Magenat, Stephane, Markus Waibel, and Antoine Beyeler. 2013. "Enki, a fast 2D robot simulator". Accessed February 18th, 2015. <http://home.gna.org/enki/>.
- Meyer, Johannes, Alexander Sendobry, Stefan Kohlbrecher, Uwe Klingauf, and Oskar Von Stryk. 2012. "Comprehensive Simulation of Quadrotor Uavs Using Ros and Gazebo." *Simulation, Modeling, and Programming for Autonomous Robots* 7628: 400-411.
- Michael, Nathan, and Vijay Kumar. 2008. "Controlling Shapes of Ensembles of Robots of Finite Size with Nonholonomic Constraints." In *Proceedings of Robotics: Science and Systems*, 41-48. <http://books.google.com/books?hl=en&lr=&id=fvCaQfBQ7qEC&oi=fnd&pg=PA41&dq=Controlling+Shapes+of+Ensembles+of+Robots+of+Finite+Size+with+Nonholonomic+Constraints&ots=VNLpIOPDoa&sig=5B3V3LCa3uMwTLaLHprfqgXe4Sw>.
- Michel, Olivier. 2004. "Webots TM: Professional Mobile Robot Simulation." *International Journal of Advanced Robotic Systems* 1 (1): 40-43.
- Mira, José. 2007. "A Multi-Robot Surveillance System Simulated in," 202-11.
- Perez, Tristan, and Thor Fossen. 2007. "Kinematic Models for Manoeuvring and Seakeeping of Marine Vessels." *Modeling, Identification and Control: A Norwegian Research Bulletin* 28 (1): 19-30. DOI: [10.4173/mic.2007.1.3](https://doi.org/10.4173/mic.2007.1.3).
- Thornton, Chris. 2006. "BugWorks - 2D Robot Simulator and Tutor". Accessed February 18th, 2015. <http://www.sussex.ac.uk/Users/christ/bugworks/>.
- Xu, Guochang. 2007. *GPS - Theory, Algorithms and Applications*. Berlin: Springer-Verlag. ISBN 978-3-540-72715-6.