

# Chapter 1

## RobotBASIC: Design, Simulate, and Deploy

**John Blankenship**  
*RobotBASIC, USA*

**Samuel Mishal**  
*RobotBASIC, USA*

### ABSTRACT

*Unlike most chapters in this book, this chapter does not introduce new methods or algorithms related to robotic navigation systems. Instead, it provides an overview of a simulation tool that, in some situations, can be useful for quickly evaluating the overall appropriateness of a wide variety of alternatives before focusing more advanced development activities on a chosen design. In addition, since the tool described herein is totally free, it can be used to help students and others new to robotics understand the value of utilizing a design-simulate-deploy approach to developing robotic behaviors. Robot Simulators can emulate nearly all aspects of a robot's functionality. Unfortunately, many programming environments that support simulation have steep learning curves and are difficult to use because of their ability to handle complex attributes such as 3D renderings and bearing friction. Fortunately, there are many situations where advanced attributes are unnecessary. When the primary goal is to quickly test the feasibility of a variety of algorithms for robotic behaviors, RobotBASIC provides an easy-to-use, economical alternative to more complex systems without sacrificing the features necessary to implement a complete design-simulate-deploy cycle. RobotBASIC's ability to simulate a variety of sensors makes it easy to quickly test the performance of various configurations in an assortment of environments. Once algorithm development is complete, the same programs used during the simulation phase of development can immediately control a real robot.*

### INTRODUCTION

The use of simulators is an indispensable and effective strategy during an engineering design process. The process of designing behavioral-control algorithms for robotic platforms can be made more effective as well as faster and easier with the use of a simulator. Nevertheless, ultimately a real robot must be used. When the simulation language also provides access to I/O ports, then the simulated algorithms and

DOI: 10.4018/978-1-5225-8060-7.ch001

programs can often be ported over to the physical robot without any changes or further programming. This allows the time and effort spent developing during simulation to become an effective and integral part of the whole design process.

Many computer languages and language libraries that support robot simulation are available. The most comprehensive offerings certainly include Matlab toolboxes and Microsoft's Robotic Studio and many others can be found with an Internet search. The ultimate choice of which system to use can be based on many factors including price, graphical rendering, available sensors, and ease-of-use. High-end programs can simulate nearly every aspect of a physical machine such as the rolling resistance of a wheel and 3D movement in realistic environments. Such detail is remarkable when needed, but such features can add tremendous complexity for users. If the primary goal is to quickly test the basic principles of a behavioral algorithm, then often a less complex tool can dramatically reduce the development time. A programming language that offers this uncomplicated approach is RobotBASIC.

## **RobotBASIC**

RobotBASIC is a full-featured, powerful, yet easy-to-use programming language with an integrated robot simulator. RobotBASIC's inbuilt functions and commands enable full control over the serial and other I/O ports thus providing the ability to directly control external hardware. Additionally RobotBASIC has a specialized protocol for seamlessly implementing wireless control over remote physical robots equipped with the appropriate firmware. Moreover, RobotBASIC may be freely used by students, universities, and researchers, even for end-user application development. It can be downloaded from [www.RobotBASIC.com](http://www.RobotBASIC.com).

The 2D implementation of the RobotBASIC simulator is deceptively simple because the real power of the design rests in the simulator's sensory capabilities. The wide range of sensors available on the simulated robot allows it to be a realistic development tool for many applications. Some of the sensors available include contact-sensing bumpers, no-contact IR object proximity detection, distance-measuring ranging sensors, a compass, line and drop-off detectors, a beacon detector, an x-y positioning GPS, and a color detecting camera. The fact that these features are fully implemented in the language and are ready to use without any user supplied programming greatly decreases development time.

## **EASY-TO-USE**

While complex programs and advanced algorithms can be implemented and tested with RobotBASIC, it is not the purpose of this chapter to explore such topics. Instead, we will utilize several unsophisticated examples to demonstrate ease-of use so that potential users can more easily determine if the system can be beneficial for them.

The example program in Figure 1 demonstrates how easily algorithms can be implemented and tested. In this example, the goal is to have the robot move to the object closest to it. The main module uses a subroutine to create an environment consisting of two randomly placed objects. The simulated robot is then initialized at the center of the environment.

9 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage:

[www.igi-global.com/chapter/robotbasic/222421](http://www.igi-global.com/chapter/robotbasic/222421)

## Related Content

---

### Emotion Generation Based on a Mismatch Theory of Emotions for Situated Agents

Clément Raïevsky and François Michaud (2009). *Handbook of Research on Synthetic Emotions and Sociable Robotics: New Applications in Affective Computing and Artificial Intelligence* (pp. 247-266).

[www.irma-international.org/chapter/emotion-generation-based-mismatch-theory/21512](http://www.irma-international.org/chapter/emotion-generation-based-mismatch-theory/21512)

### Edge AI for Real-Time and Intelligent Agriculture

Jigna Bhupendra Prajapati, Akash Kumar, Jhila Pramanik, Bhupendra G. Prajapati and Kavita Saini (2023). *Applying Drone Technologies and Robotics for Agricultural Sustainability* (pp. 215-244).

[www.irma-international.org/chapter/edge-ai-for-real-time-and-intelligent-agriculture/317075](http://www.irma-international.org/chapter/edge-ai-for-real-time-and-intelligent-agriculture/317075)

### Aerodynamic Analysis of Supersonic Spikes for Drag Reduction

Md Akhtar Khan and K. Vigneshwar (2023). *Global Perspectives on Robotics and Autonomous Systems: Development and Applications* (pp. 130-167).

[www.irma-international.org/chapter/aerodynamic-analysis-of-supersonic-spikes-for-drag-reduction/327571](http://www.irma-international.org/chapter/aerodynamic-analysis-of-supersonic-spikes-for-drag-reduction/327571)

### Epistemology and Emotions

Jordi Vallverdú (2013). *International Journal of Synthetic Emotions* (pp. 92-94).

[www.irma-international.org/article/epistemology-emotions/77657](http://www.irma-international.org/article/epistemology-emotions/77657)

### Robotics in Surgical Techniques Robotics in Surgical Techniques: Present and Future Trends

Vassilia Costarides, Apollon Zygomas, Kostas Giokas and Dimitris Koutsouris (2019). *Rapid Automation: Concepts, Methodologies, Tools, and Applications* (pp. 1528-1542).

[www.irma-international.org/chapter/robotics-in-surgical-techniques-robotics-in-surgical-techniques/222498](http://www.irma-international.org/chapter/robotics-in-surgical-techniques-robotics-in-surgical-techniques/222498)