Windows-based Robot Simulation Tools

Zafer Bingül
Kocaeli University,
Department of Mechatronics Engineering,
Kocaeli, Turkey,
Email: zaferb@kou.edu.tr

Poolsak Koseeyaporn and George E. Cook
Vanderbilt University,
Department of Electrical Engineering,
Nashville, Tennessee, USA,
Email: pks@kmitnb.ac.th,
george.e.cook@vanderbilt.edu

Abstract

A new graphical simulation package, which is capable of interoperability with other extensive programs, is proposed. A built-in LISP interpreter based on COMMON LISP allows a user to model and manipulate the robot while the result is displayed on a three-dimension graphical display. Other programming languages that are compatible with the Component Object Model (COM) interface can also be applied to the modeling and simulation. The simulation package provides many useful features such as collision detection, inverse kinematics, matrix transformation operations, and several graphical user interfaces. The simulation objects are built based on a triangle strip model to form primitive objects such as a box, cone, sphere, etc. These primitive objects are combined to construct the robot that may contain several serial or parallel links so that many possible types of robots can be modeled.

1 Introduction

The robot simulation package proposed here is developed from a versatile simulation program called ROBOSIM. ROBOSIM is an extensible robot simulation program that is used to model and simulate robots on a low cost PC platform. The current version developed by James F. Springfield [1] has been used extensively in research and robotics courses [2]. From an educational perspective, ROBOSIM has successfully satisfied many learning objectives such as modeling and transformation of rigid structures, forward and inverse kinematics computation, and path planning with collision avoidance. Many types of robots containing either serial or parallel links can be modeled or simulated in this program. The user can model or manipulate robots by employing the COMMON LISP programming language. Each serial or parallel link of a robot can be formed from primitive objects such as a cone, sphere, box, cylinder or more complex geometrical objects such as extruded surfaces or revolved surfaces. Therefore, many possible types of robots can be modeled and simulated. By either loading from a LISP file or typing LISP commands in an interactive command line interface, the user can execute each command to model or simulate robot motions. The forward and inverse kinematics programming allows the user to gain increased understanding of the accurate position or motion control of the robot. Collision detection can be performed by either applying an exception handling procedure or a collision check command. This allows performing artificial intelligent path planning as used in manufacturing processes. ROBOSIM also supports extensible routines and graphical user interface (GUI) programming under the COMMON LISP programming language.

However, there are some limitations of extending features of this simulator. It can only be extended under the COMMON LISP programming language, and there are only a few GUI facilities provided [1]. Furthermore, it requires knowledge of previous implementations in order to develop or extend the program since the new implementation cannot be created as a separate process. Dynamic simulation is another feature that has not been included in the current version of ROBOSIM. These limitations can be overcome by employing the Component Object Model (COM) technology [3] that has the characteristics of reusability, extendibility and language independence. Because ROBOSIM has been implemented in the C++ programming language under the Win32 platform, COM that can be implemented from the C++ programming language is an appropriate approach to implement this simulation program. It has been shown in [4] that this programming strategy can also be applied to a manufacturing robot and has more flexibility as compared to other architectures. With it, users can extensively implement their own effective graphic user interface as in [5] to model and simulate robots without requiring knowledge inside of the ROBOSIM implementation. These extended programs can be implemented from many different programming languages that support COM, for instance, Visual Basic, Visual C++, etc. Simulating robot dynamics is another feature that can be achieved by linking the robot simulation components with Matlab SIMULINK [6]. One more benefit of using COM to develop this simulation program is that distributing the program via the Internet can be done without much more effort. This is another useful feature since the Internet is popular for robot simulation research and the network is becoming faster, cheaper, and growing rapidly [7-10]. The details