

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900

Open access books available

185,000

International authors and editors

200M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



Physical Interaction and Control of Robotic Systems Using Hardware-in-the-Loop Simulation

Senthil Kumar Jagatheesa Perumal and Sivasankar Ganesan

Abstract

Robotic systems used in industries and other complex applications need huge investment, and testing of them under robust conditions are highly challenging. Controlling and testing of such systems can be done with ease with the support of hardware-in-the-loop (HIL) simulation technique and it saves lot of time and resources. The chapter deals on the various interaction methods of robotic systems with physical environments using tactile, force, and vision sensors. It also discusses about the usage of hardware-in-the-loop technique for testing of grasp and task control algorithms in the model of robotic systems. The chapter also elaborates on usage of hardware and software platforms for implementing the control algorithms for performing physical interaction. Finally, the chapter summarizes with the case study of HIL implementation of the control algorithms in Texas Instruments (TI) C2000 microcontroller, interacting with model of Kuka's youBot Mobile Manipulator. The mathematical model is developed using MATLAB software and the virtual animation setup of the robot is developed using the Virtual Robot Experimentation Platform (V-REP) robot simulator. By actuating the Kuka's youBot mobile manipulator in the V-REP tool, it is observed to produce a tracking accuracy of 92% for physical interaction and object handling tasks.

Keywords: hardware-in-the-loop, control algorithms, robotic manipulators, mobile robots, physical interaction

1. Introduction

Robotics have been flourished as a platform with its extensive set of applications, which once was thought as a night mare, and now have started dominating the industrial sector. Because of its expanded domain of applications, it can even start ruling humans in near future. Robotic systems have been turned into very powerful essentials of today's industrial sector. It is often anticipated to synthesis certain features of human tasks in huge scale by using actuators, sensors, and personal computers or on-board computers. Designing and controlling of such robotic systems requires combining the concepts from various classical fields of science [1].

Robotic manipulators are a class of robotic systems with mechanical-articulated arm along with a static base. On the other hand, mobile robots belong to another different class of robotic systems with mobile base. Currently, most of the automation tasks in industries also depend on hybrid variant of robotic systems, which

have both manipulator and mobile base [2]. Usage of robots mainly aims to reduce man power in control appliances for industrial and other household or commercial appliances, by enabling accuracy in processing and performing physical interactions. Actions performed by robotic systems are relatively faster than humans and also deliver reliability and robustness in the system.

Various robotics research domains are available with their potential paths that work together with human beings in households and workplaces as useful and capable agents based on the ability of robots. Another most essential objective of robotic systems are interacting with the physical environment toward specific applications domains, tracking the objects, and moving the gripper with accuracy [3]. Also precision in trajectory tracking of robotic systems, grasping of objects, stabilizing the grip and force control are all challenging research issues, which in turn may seriously affect the production in industrial sector if they are not properly addressed [4].

For performing physical interaction with the help of robotic systems, handling of objects in industries and other applications are becoming predominant. It is driven by proper motion control accomplished by fixing global, joint, and object's reference frames in the physical environment [5]. The choice of workspace of the robotic systems also plays a major role for physical interaction. Workspaces are the collection of points that can be reached by the robot based on its configuration, size of links, type of joints, and also defined by its own limitations.

For establishing the physical link of the robot with the environment, tactile sensors along with force sensors and vision sensors can be deployed based on the variety of applications. Sensing of the handled objects and force applied on the particular object based on dexterous capability of the objects are vital parameters in physical interaction. Controlling the force applied on the objects and stabilizing the grasping potential of the grippers are crucial challenges while performing interaction with the environment. Various controllers were proposed by researchers for reducing the control effort with reduced noise in control torque. Also for operating the robotic systems under unknown external environments, certain adaptive techniques were also proposed by researchers. Accuracy in driving of the end-effectors of the robots can be accomplished by tracking the desired path for each joint of robots by choosing the best closed loop controllers. In this chapter, systematic solutions are provided for the design of grasp and force control along with the support of visual servoing for controlling the hybrid robotic systems with the physical environment [6].

2. Physical interaction of robotic systems

Robotic systems can interact physically in its workspace by first establishing a contact with the environment with appropriate configuration. Subsequently, the robots are fed with required force for motion of its physical parts to perform certain desired tasks. Interaction by means of grasping of objects in workspace and the desired motion control constrained to tasks are dealt. Since, the early inception of robotic research, manipulation of robots with its environment has been dealt by different researchers.

2.1 Interaction with environments

Robotic systems grasping objects in the environment can be implemented based on either contact level method or knowledge-based method. Contact level method applies force torque to the objects handled by the robot. Few of the variants of the contact-based methods include frictionless contacts, soft contact, and with

frictional contact forces applied. Reliability of grasping potential can be estimated using force closure property, by compensating external disturbances. Other possible forces of contact can also be random potential grasps, which may not be optimal but manages vector space that includes all feasible contact forces.

The limitation of contact-based approach is that it never considers the constraints imposed on the hand of the robotic systems. This may result in target points that may not be reachable by the hand. Knowledge-based method considers the hand post postures that are already predefined. This provides a qualitative method for planning for grasping objects by adapting to the workspace geometry. Control algorithms based on task-oriented grasping consider requirements of tasks in the workspace of the robots. This method requires minimum number of contacts than other conventional methods. So, the number of fingers in the workspace of the robotic systems can be minimized. For any given task, the efficiency of task-oriented approach is reasonably better, and it also increases the versatility of the robot for performing different tasks.

2.2 Role of sensors for physical interaction

Usage of tactile sensors are one of the best choice for performing physical interactions of robots with the environment. They are also used for estimating the contact information and detecting the slippery of objects during grasping. Force sensors on the other hand estimates the impact of forces applied on the objects and studies the dexterous capability of the object under test. Vision sensors are one of the best choices for observing the impact of physical interaction of the robot with the environment [7]. However, it also challenges the designer with series of preprocessing stages during implementation. Model-based pose estimation techniques can simultaneously observe the position as well the orientation of the hand and the objects under manipulation.

Best and robust estimation of environmental parameters during physical interaction of the robotic systems can be availed by implementing sensor fusion techniques. Sensor fusion provides mechanisms for combining different sensor outputs for better observation about the environment. Accurate prediction of environmental parameters is possible only if the information acquired from multiple sensors is combined for decision making. Popular approaches such as Kalman Filter and Extended Kalman Filter can be better choices for sensor fusion processes, particularly in robotic applications.

2.3 Planner for physical interaction

The requirements of physical interaction and tasks can be defined with the support of interaction planning algorithms. Physical interaction tasks can be automatically specified, by feeding the description of objects and tasks to be handled at workspace to planning algorithms. Choice of ideal gripper, gripper pre-shapes, and gripper adaptors provides generic task-based planning algorithms for the end-effectors of robots with good versatility. Also, the planning algorithm may also be environmental specific based on the dexterous capability of the objects for grasping and tasks performed on the environment.

Vast category of objects may limit the physical interaction tasks supported by the planner. Coupling the planner with vision-based sensors for physical interaction with strange objects in the environment provides perfect versatility and autonomy for the robotic systems. Approximating the shapes of the strange objects, the planning algorithms generate suitable velocity and force references for the end-effectors to actuate the joint actuators.

2.4 Application of force and feedback during interaction

During physical interaction of the robot with the environment, even a minimum change in the positioning may lead to generate unwanted interaction forces by the planner. Choice of active and passive stiffness methods can be employed to regulate the application of the forces on the end-effectors. Based on the requirements of task, the active stiffness method controls the desired end-effector stiffness. Passive method enables to deform the mechanical body of the gripper based on the external forces applied.

3. Control of physical interaction

During physical interaction using the robot's gripper with the objects, usage of tactile sensors enables to acquire the contact force and pressure information. Forces from the hand can be observed by placing a force sensor in the wrist of the robot [8]. It ensures to give force feedback always, whenever some kind of forces at any part of the hand is sensed. Force feedback is a vital observation in physical interaction to detect jointly the object contact at the end effector and the motion of the hand simultaneously. **Figure 1** shows the generic control scheme imparted for physical interaction using force and tactile sensors for feedback. For the improved manipulation, vision-based sensors can be combined with tactile sensors.

3.1 Grasp and force control for object positioning and handling

To stipulate few dynamic behaviors of robot in its environmental workspace, its contact with the objects can be controlled using impedance control method coupled with active stiffness. Position of the gripper with respect to the contact force can be obtained using reconfigurable mechanical impedance in the workspace. It can be derived based on second order transfer function equivalent to mass spring damper. **Figure 2** shows how the desired impedance can be used to modify the position by generating appropriate position control signals. The impedance transfer function generates a position error with respect to the difference between the reference force and the force feedback from the robot. The position controller uses the error feedback computed from the reference position and current Cartesian position of the robot, to generate the control signal for moving the robot's gripper actuators.

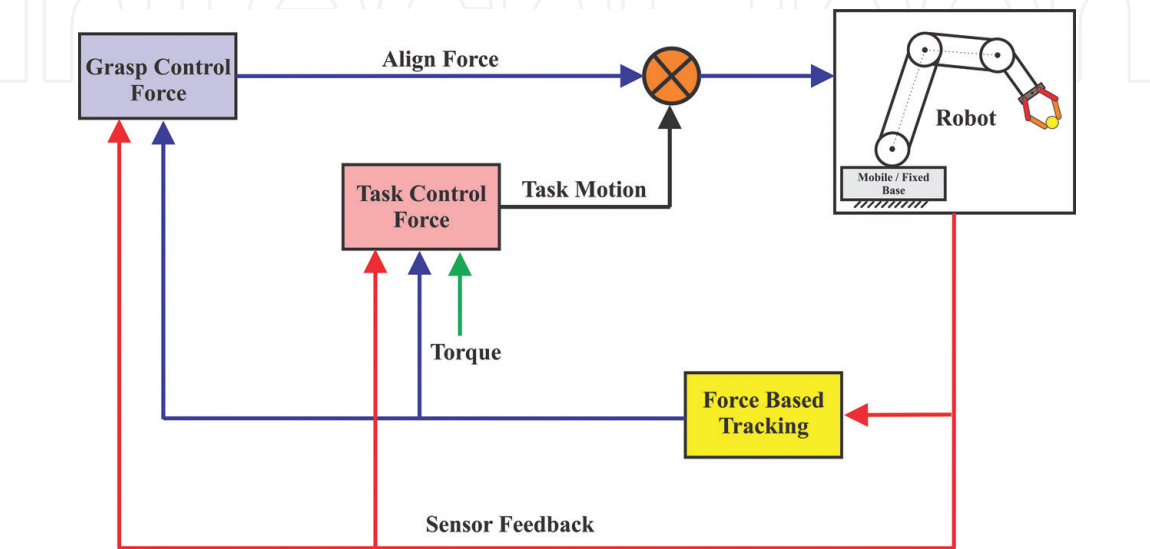


Figure 1.
The general force control mechanism for physical interaction of the robots.

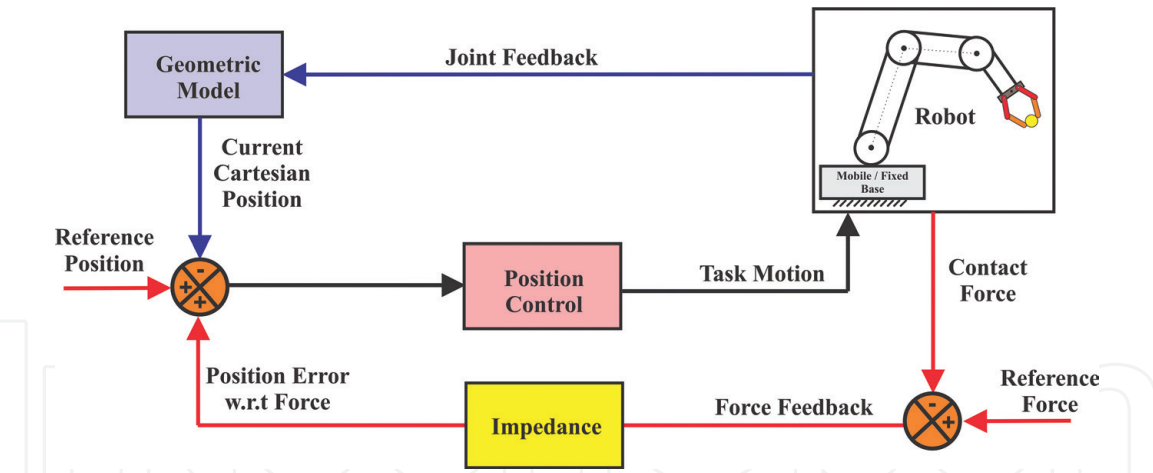


Figure 2.
General control approach of the position based on impedance.

The interaction forces and its control can be performed using indirect method by controlling the contact forces explicitly or by implementing hybrid techniques. Those hybrid approaches combine the force control and position control on the same directions. **Figure 3** shows the generic method of the hybrid position force control by including position and force error filters. A frame fixed to the task handling part of the end-effector can be specified using a matrix in which the value of 1 represents axis of force control direction and value of 0 indicates the axis of position control. This particular approach very spontaneously allows the control law implementation for the physical interaction tasks. With precise knowledge of the frame and the environment, it ensures that no disturbance appears between the directions of position and force [9]. If the physical interaction is deployed on unstructured type of environment, this kind of control will be quite challenging. This approach utilizes a position error filter and force error filter, which filters the unwanted errors to drive the position and force controller. Generated hybrid signal from the deployed control law drives the gripper actuator for task handling.

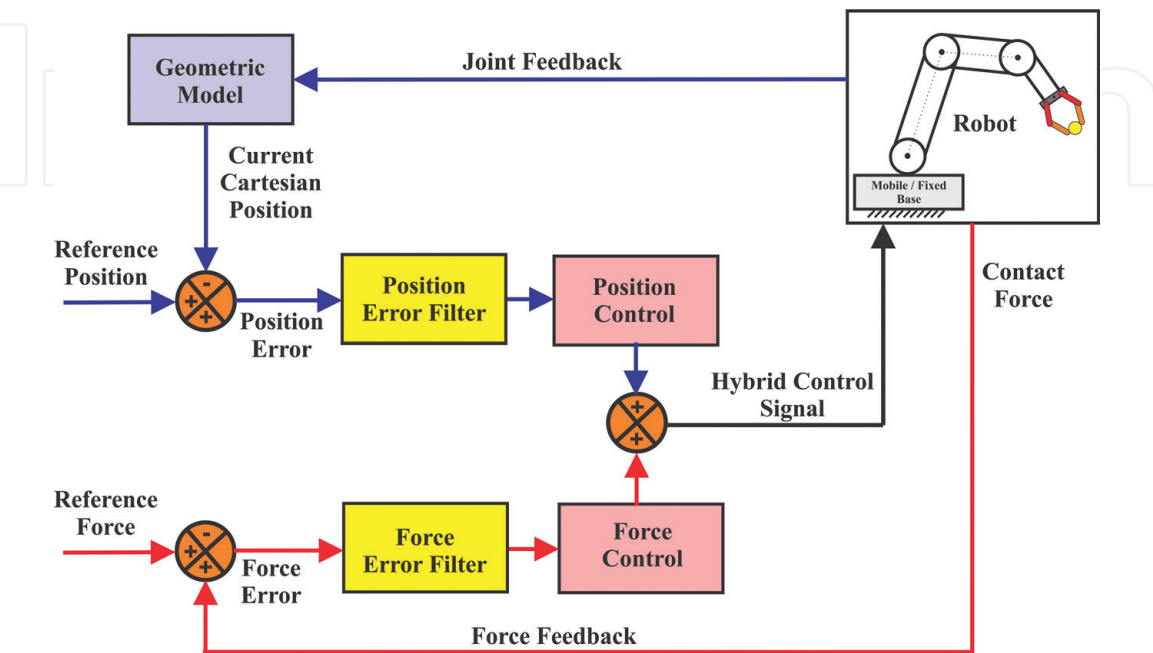


Figure 3.
General approach of the hybrid position force control.

Simultaneous control of position and force along the same direction can be achieved with the support of hybrid external position force control in physical interactions tasks. In this architecture, the inner position control loop will be driven by an outer force control loop as shown in **Figure 4**. The new reference position is computed from the force control signal estimated through the reference force and feedback force error. The force error is added to the existing reference of the position [10]. This provides simplicity in the architecture and implemented above the existing position controller of the robot. The hybrid control law is finally based on the position control signal generated in the control system using computed position error [11].

3.2 Visual control for object positioning and handling

Vision-based sensors play a most vital role during the physical interaction phase. When the robot does not interact with the environment, information from force and tactile sensors will not be available. In such circumstances, vision sensor data can be acquired in order to trace the target object in the environment. This enables to guide the robot's hand toward the object in the robot's workspace. For grasping the objects in the workspace, pose estimation methods can offer an approximate position and orientation of the target object. The adopted algorithm complexity is of polynomial time, for which the accuracy varies with the chosen optimization techniques [12]. The physical look-based pose estimation method does not need familiarity about the 3D model of the object. In model-based methods, the pose accuracy is better than physical look-based pose estimation, but it usually needs a suitable initial estimate in addition to the model of the object. The task programmer has to choose the most suitable approach depending on the available previous knowledge.

During physical interaction phase, vision-based pose estimation algorithms can be used to investigate the properties of the physical interaction on the object. When the robotic system executes the task motion, desired vital measures can be detected, such as the angle of gripper opening, reachability of the object, failure of grasping, and many more. However, the chief impact of vision sensors during physical interaction tasks, is that it is promising to track the progress of the frames fixed in the environment. This can be accomplished by a direct observation on the hand and the object in the workspace. This completely avoids the necessity of force-based techniques for object tracking. Grasping of objects can be detected directly with

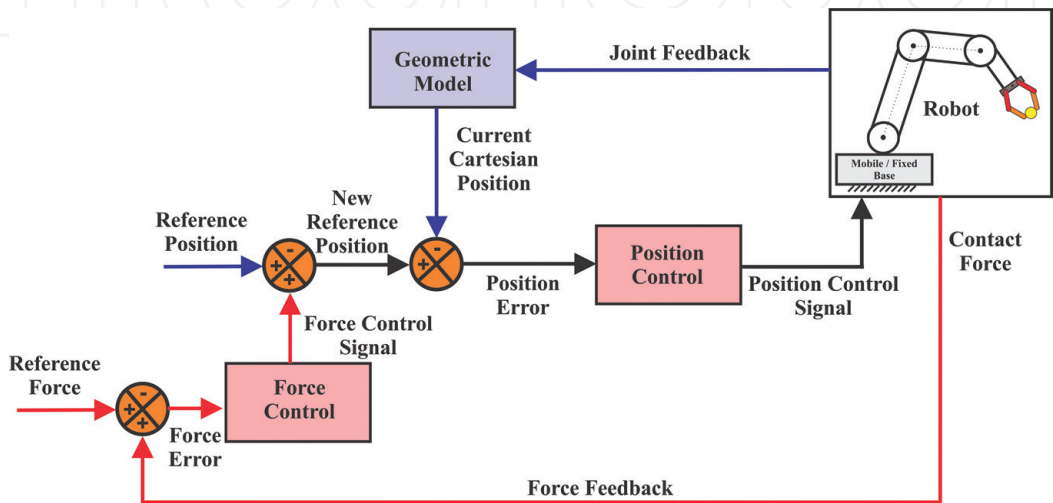


Figure 4.
General approach of the hybrid external position and force control.

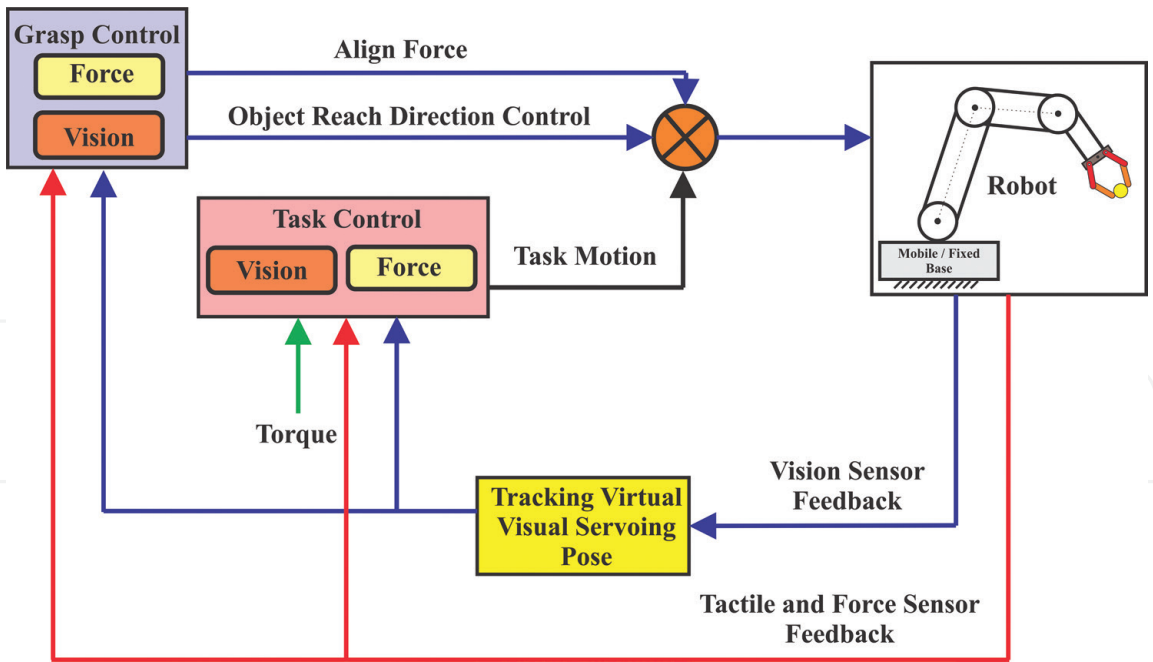


Figure 5.
The general vision and force control scheme for the physical interaction.

the aid of vision sensors, by detecting a specific sequence, without formulation of a predefined path for the arm and gripper. It is absolutely mandatory to perform sensor fusion of the visual signal with force feedback, to manage unforeseen forces due to vision sensor calibration and preprocessing errors.

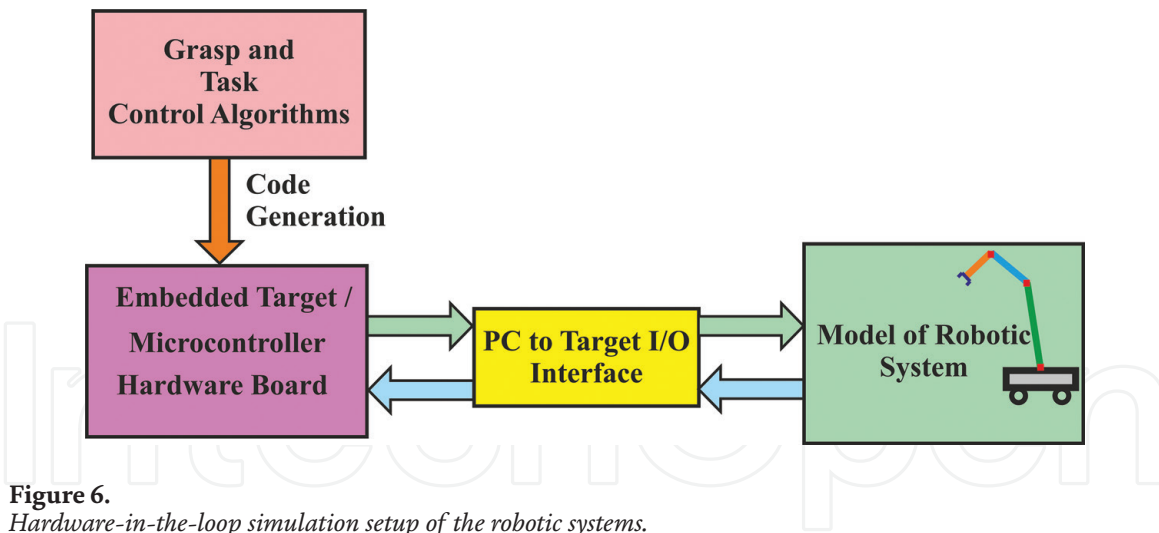
Figure 5 outlines the main impact of vision sensors combined with tactile and force sensors for grasping objects during physical interaction. They together can trace the objects in the physical environment, thus empowering accurate reaching and alignment between the object and robotic hand, before and after physical contact. After physical contact, alignment task needs to be performed with the support of force controller, in order to consistently handle and stabilize the misalignments caused between object and hand due to deployment of vision sensors.

4. Hardware-in-the-loop simulation

Certain physical systems are prone to possess complex design, unsafe to test in real time, subjected to operate in certain real time systems, and not economical. Robots also belong to one such category of physical systems. For such robotic systems, development of the embedded control system and testing are made feasible with the support of HIL simulation technique [1].

As a robotic system design requires multidisciplinary mastering, partitioning the design tasks into various subsystems simplifies their analysis and synthesis. So, by utilizing the real hardware modules in the loop of real time simulation enables the detailed analysis of sensor noises and actuator limitations of robotic systems [2]. This can be accomplished by HIL technique, where the control algorithms are implemented in the actual hardware, and it controls the simulated model of the robotic system.

In the adopted HIL methodology, complexity of the robots to be controlled is modeled by including all its related dynamics by equivalent mathematical representation of the systems included in test and development. The embedded target runs the control algorithm for the control of joint actuators and they interact with the simulated model of the robot. **Figure 6** shows the HIL simulation setup of the



robotic systems [3]. It is implemented with the control algorithms running on the embedded target board. The board is interfaced to the PC, which runs a model of robotic system via Input/Output interface. Certain modeling tools possess simulation of the complex mechanisms in the robotic systems.

The usage of HIL simulation techniques increases the safety in operating the robots and enhances the production quality after analyzing the performance in HIL platform and later on switching to actual physical robotic system. Another striking feature of using HIL is that it saves time and money to a larger extent.

4.1 Steps in HIL simulation

The core steps in HIL simulation of the robotic system carried out are as follows:

1. Developing the mathematical model of the actual physical robotic systems using the tools such as MATLAB, LabVIEW, or other open source platforms. It must also be ensured that the developed model can be accessed by the hardware devices with appropriate communication.
2. Developing the control algorithms for robotic systems using the open source or proprietary tools for the core embedded target board. It acts as a heart of the system running the task and grasp control algorithms.
3. Configuration of the environment variables in the software environment based on the appropriate compiler choice, bios setup, device support form control suite, and flashing utilities, etc.
4. The target preferences in the embedded coder of software environment need to be configured by choosing the appropriate target processor by mapping the developed algorithms to the input, output pins of the controller.
5. Perform the HIL simulation by testing the implemented control algorithms on the embedded target board along with simulated mathematical model of the robotic system.

These sets of tasks investigate the crucial awareness of performing the HIL simulation using embedded target controller, modeling, and simulation tools. Initial step on this process is to work on the virtual model of the simulated plant under test

using the software tool. Followed by, testing of the implemented control algorithms running in the embedded hardware on the simulated model of the robotic system is performed. Finally, these test analyses give the confidence for driving the actual robotic system using the grasp and task control algorithms.

5. Case study using Kuka's youBot mobile manipulator

The KUKA's youBot is a mobile robot manipulator that was designed targeting the scientific research and training community as an open source platform. **Figure 7** shows the picture of KUKA's youBot mobile manipulator with a mobile base and manipulator mounted on top of it [13]. Since, youBot can be used for development and validation of mobile robot platform as well as for fixed manipulation tasks, youBot is considered in our case study for physical interaction with the environment.

5.1 Specifications of KUKA's youBot

KUKA's youBot consists of an omnidirectional mobile platform along with 5-degree of freedom (DOF) manipulator that possess two finger gripper, which can be controlled independently. The omnidirectional base of the robot has a payload of 20 kg and it is driven by four omni wheels, which allows the movement of wheels in all directions without any external mechanical steering. Each wheel consists of a series of rollers mounted at a 45° angle. The wheels are driven by brushless DC motors with built-in gearbox, relative encoder and joint bearing. Wheel motion is controlled independently through drives using commands. The lowest level of command being pulse width modulation (PWM) and higher level control commands are: current control (CC), velocity control (VC), and position control (PC).

Youbot's arm can manipulate a load of up to 0.5 kg in any position in three dimensions. The distributed axis controllers are connected via the EtherCAT backplane bus EBUS. Each joint in youbot arm has its own servo controller, which contains: ARM Cortex-M3 microcontroller, hall sensors, EtherCAT interface, position, velocity, and current PID-controllers. Using predefined positions provided



Figure 7.
KUKA's youBot mobile manipulator (image courtesy: <http://www.youbot-store.com/>).

over WLAN or Ethernet as input, the PC calculates moment, speed or position instructions for each axis and handles their interpolations individually.

Youbot's arm wrist is equipped with a two finger parallel gripper with a 20 mm stroke. There are multiple mounting points for the gripper fingers that user can choose based on the size of the objects to pick up [14]. In the standard gripper, the jaws are operated by two stepper motors.

Operation of the robot is possible with both connected power supply unit and the batteries. The power controller features separate charging controls for the two maintenance-free lead-acid batteries (24 V, 5 Ah) when the charger (200 W) is connected. If no charger is connected, the two batteries can supply power for up to 90 minutes. In addition, the power controller features individually switchable power supplies for the computer and the motors. Battery current is monitored via the on-board computer.

5.2 V-REP and MATLAB interface for youBot

V-REP is an open source robot simulation tool that possesses various features, relatively independent functions and more elaborate application program interfaces (APIs). It is one of the stable platforms than other open source tools with easy plugin and configuration of robotic systems [15]. Each object/model in V-REP scene can be individually controlled via an embedded script, a plugin, a Robot Operating System (ROS) node, a remote API client, or a custom solution. Control algorithms can be written in C/C++, Python, Java, Lua, MATLAB, and Octave or Urbi.

Remote API can be implemented by blocking function calls, non-blocking function calls, data streaming, and synchronous operation. In this research, we used MATLAB as the remote API because it provides a very convenient and easy way to write, modify, and run. This also allows controlling a simulation or a model with the exact same code as the one that runs the real robot. The remote API functionality relies on the remote API plugin and the remote API code on the client side. Both programs are open source and can be found in the “programming” directory of V-REP’s installation. Robot HIL control system is connected with V-REP robot simulator through MATLAB remote API function as shown in **Figure 8**.

Simulation scene in V-REP robot simulator contains several elemental objects that are assembled in a tree-like hierarchy and operate in conjunction with each other to achieve physical interactions. In addition, V-REP also possesses several calculation modules that can directly operate on one or several objects in a scene. Major objects and modules used in the simulation scene include (i) sensors, (ii) CAD models of the plant and robot manipulator, (iii) inverse kinematics, (iv) minimum distance calculation, (v) collision detection, (vi) path planning, and (vii) visual servo control. Other objects that were used as basic building blocks are: dummies, joints, shapes, graphs, paths, lights, and cameras.

V-REP supports different vision sensors (orthographic and perspective type) and proximity sensors (ray-type, pyramid-type, cylinder-type, disk-type, and

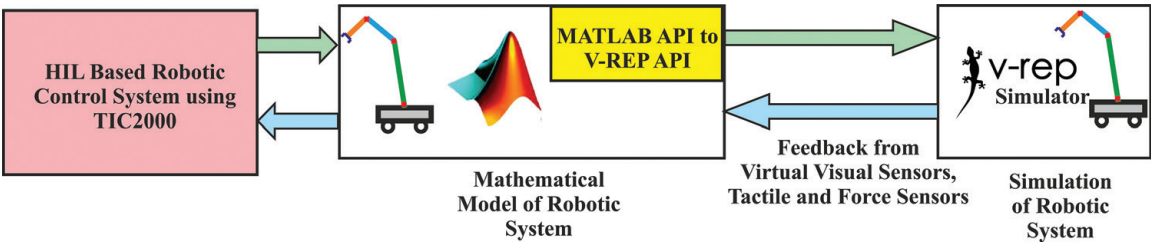


Figure 8. Schematic diagram for robot HIL system with V-rep through MATLAB API.

cone- or randomized ray-type proximity sensors). In this study, we used tactile sensors, force sensors, camera, RGB sensor, XYZ sensor, and laser range finder.

Following are the steps to be followed for interfacing V-REP with MATLAB.

1. Load the virtual model of Kuka's youBot in to V-REP environment by selecting youbot from the mobile robot model available in Model browser tab.
2. To enable control from MATLAB API, call the function `simRemoteApi.start` (port address) with specific port address in the V-REP script.
3. To establish link to the MATLAB API calls, place the two supporting MATLAB files (`remApi` and `remoteApiProto`) and `remoteApi.dll` file to your currently working folder. Those files can be located in V-REP's installation directory, in the directory `programming/remoteApiBindings/matlab`. Also make sure that MATLAB uses the same bit-architecture as the `remoteApi` library: 64bit MATLAB needs 64bit `remoteApi` library.
4. Then run the MATLAB script code ensuring that V-REP simulator is already running.
5. Observe the animation of the virtual model of Kuka's youBot in to V-REP environment for object handling based on the control signals from the C2000 microcontroller.

5.3 HIL-based control law implementation

Commands to the actual physical youBot drives are sent from an Intel Atom on-board computer running on a real-time Linux kernel for the Simple Open EtherCAT Master (SOEM). A real-time communication is established between drives and on-board computer using EtherCAT, a technology used in KUKA's industrial robots.

The KUKA youBot drive protocol is open source, which encourages the users to develop their own applications and control systems. This flexible feature enables us to deploy the control algorithm in Texas Instruments C2000 microcontroller. Since, we deploy HIL-based control law and virtual model of the youBot, we eliminate the Intel Atom on-board system and use the TI C2000 hardware target board for deploying the grasp and task control algorithms. **Figure 9** shows the HIL setup for the robotic systems using TI C2000 real time controller, which runs the vision and force control scheme for the physical interaction along with the model of the youBot developed using MATLAB. The animated actions of the youBot are observed using V-REP robot simulator for object handling controlled using the HIL setup.

MATLAB Simulink environment has the support for deploying the developed algorithm, model of environment in the real time target boards, thereby providing support for HIL. MATLAB also possess the capability to interface Texas Instruments C2000 microcontroller board. With those extended support, the control algorithms are deployed in the C2000 microcontroller board.

For carrying out those tasks, following steps are done:

1. First, the kinematic and dynamic model of Kuka YouBot mobile manipulator was developed in the MATLAB Simulink Environment.
2. Control algorithms for physical interaction of the robot for prescribed environmental characteristics are developed using MATLAB Simulink Environment.

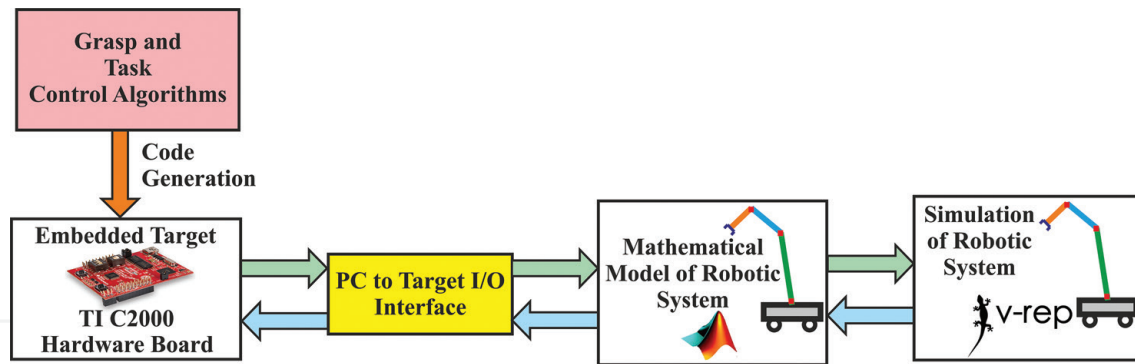


Figure 9.
Hardware-in-the-loop simulation setup of the robotic system.

3. Executable model of the control algorithms is downloaded in the C2000 microcontroller board for building a HIL simulation platform.
4. The control algorithms run on the hardware board, which replaces the developed software.
5. Testing of the model is done in conjugation with the developed model of the Kuka YouBot mobile manipulator.

Repeated testing of the control algorithms are performed until meeting desired performance. Steps 4 and 5 are repeated for gaining confidence of real time implementation of the control algorithms in the physical Kuka YouBot mobile manipulator.

These testing phases enable to deploy the algorithm in real time. This was demonstrated using the model of Kuka YouBot mobile manipulator in V-REP robot simulation environment. Sensors data are taken from the V-REP environment as input to the HIL systems. Performance of the robot was assessed for the implemented control algorithms and the results are presented as shown in **Figures 10–12**.

5.4 Physical interaction of youBOT

Commands to the actual physical youBot drives are sent from an Intel Atom on-board computer running on a real-time Linux kernel for the Simple Open EtherCAT Master (SOEM). A real-time communication is established between drives and on-board computer using EtherCAT, a technology used in KUKA's industrial robots. The KUKA youBot drive protocol is open source, which encourages the users to develop their own applications and control systems. This flexible feature enables us to deploy the control algorithm in Texas Instruments C2000 microcontroller. Since, we deploy HIL-based control law and virtual model of the youBot, we eliminate the Intel Atom on-board system and use the TI C2000 hardware target board for deploying the grasp and task control algorithms.

Figure 10 shows the snapshot of the animated Kuka's youBot performing physical interaction with the objects in the environment simulated using V-REP robot simulator. The trajectory taken by the robotic arm, and the force exposed for grasping of objects are commands from the HIL-based robot control algorithms implemented using TI C2000 microcontroller board. The force and tactile feedbacks along with visual feedback are acquired from robot deployed in the V-REP environment. Those signals are given as an input to the control algorithms executing in the C2000 controller through the MATLAB environment.

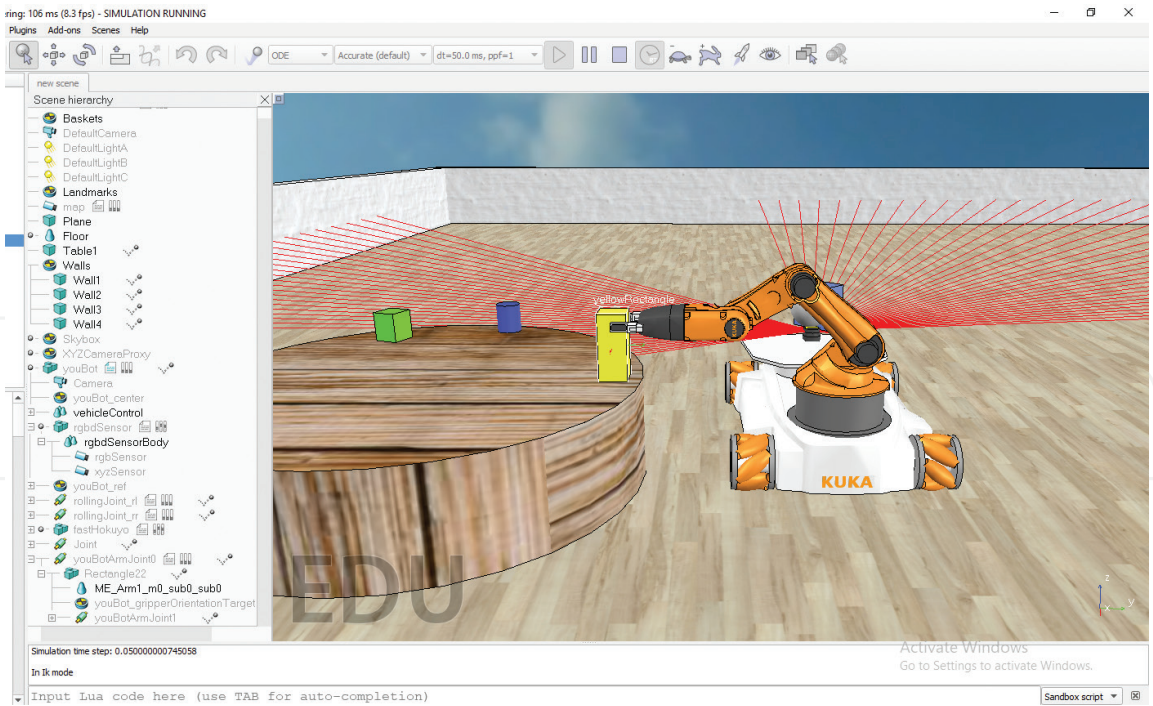


Figure 10.
Kuka's youBot performing physical interaction with the objects in the environment simulated using V-REP robot simulator.

Range of sensors monitoring the environment during the physical interaction of the robot, gives the acquired environmental parameters to the MATLAB environment through the APIs. **Figure 11** shows the 2D visual range of the youBot feedback along x and y axis from the virtual physical environment acquired from the V-REP robot simulator.

Based on the acquired information and additional sensor feedback to the MATLAB environment, sensor fusion is done using Kalman filter. The probabilistic range of output of Kalman filter, data are fed to the C2000 real time controller through the target PC connector. The vision and force control algorithm running on the C2000 board, generate control signals based on the required tasks and grasp forces for dynamic range of objects in the physical environment. This process ensures to give the required force, maintains the stability of the grasped objects, and avoids slippery of the objects in the robot's hand. Initially, the tracking characteristics of the youbot are tested for a circular trajectory for object handling. From **Figure 12**, the plot shows the deviation of the actual trajectory from the desired

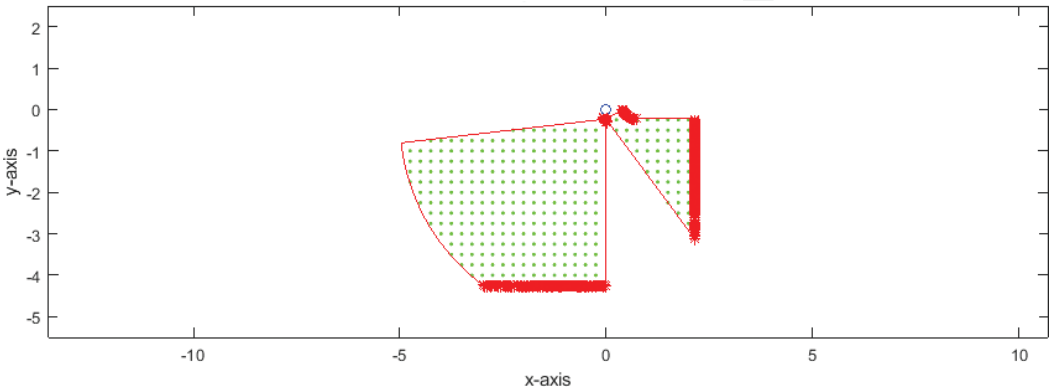


Figure 11.
2D visual range of Kuka's youBot feedback to the MATLAB environment from V-REP along x and y axis.

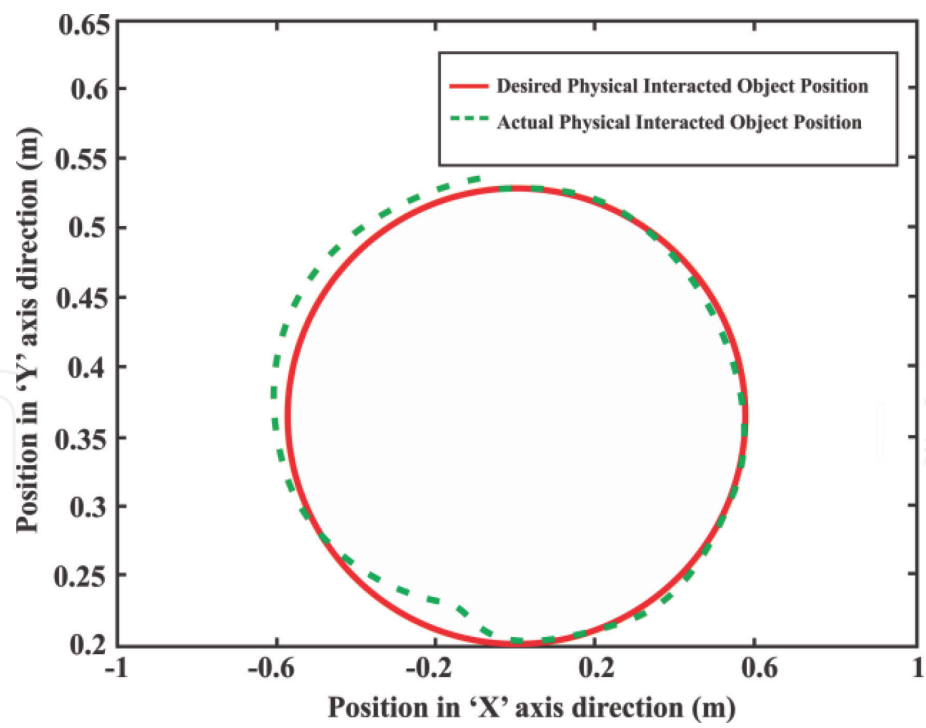


Figure 12.
Circular trajectory tracking characteristics of Kuka's youBot for the vision and force control algorithm implemented in TI C2000 real time controller.

trajectory. For different observations, it was evident that the deviation is around 8% for the proposed algorithm.

Based on the satisfactory performance observed from tracking the circular trajectory, the youbot was configured with the application toward object handling. It was engaged in pick and place of an object using the proposed vision and force control algorithm implemented as in the HIL setup shown in **Figure 6** using TI C2000 real time controller. **Figure 13** shows the tracking of desired trajectory

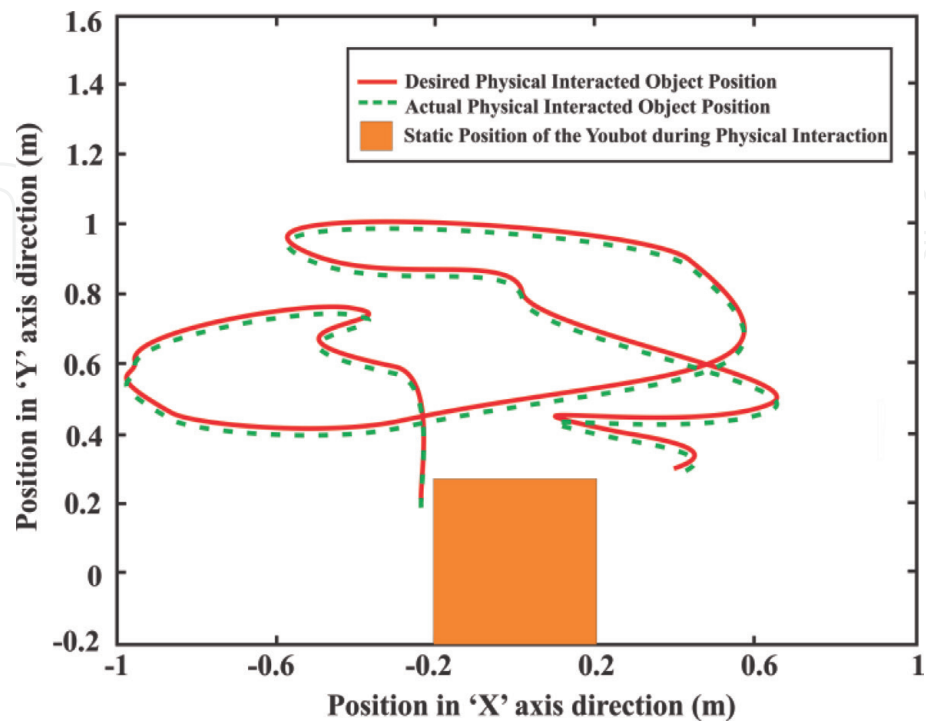


Figure 13.
Object handling trajectory of static Kuka's youBot for the vision and force control algorithm implemented in TI C2000 real time controller.

during the physical interaction of the youbot with the environment. It is observed that the youbot in a static position engaged in physical interaction with the objects. The actual trajectory is deviated by 8% from the desired trajectory for the physical interaction of objects in the environment. This ensures better tracking accuracy of 92% in the implemented control algorithm using the proposed HIL setup and TI C2000 hardware for robotic applications. Further improvement can be ensured with the choice of appropriate adaptive sensor fusion techniques, control algorithms, and optimization techniques for fine tuning the system parameters.

6. Conclusion

The goal of this chapter is to implement HIL-based grasp and task control algorithms supported by using the vision-based sensors, tactile, and force sensors. The objective was to test the performance of the youBot arm for physical interaction with an environment and control its manipulation using the algorithms implemented in C2000 real time controller based on HIL simulation technique. HIL-based control algorithms actuate the joints of virtual youBot model implemented with appropriate scenario in V-REP robot simulator driven by the model of the robot implemented using MATLAB. After the analysis of performance of HIL-based grasp and task control algorithms youBot arm and a study of the pre-existing control and hardware, we proposed a new decentralized control architecture. The control algorithms are tested first using C2000 controller on a V-REP robot simulator, then by observing satisfactory performance of the algorithms they can be ported to real youBot arm. The HIL-based results of the vision and force controller give confidence level of 92% to deploy them on real youBot mobile manipulator. It can be further improved with more adaptive and optimization techniques.

Acknowledgements

The authors gratefully acknowledge the support of the Management of Mepco Schlenk Engineering College, Sivakasi, Tamil Nadu, India, the Principal and Head of Electronics and Communication Engineering Department for their constant support and encouragement.

Author details

Senthil Kumar Jagatheesa Perumal* and Sivasankar Ganesan
Department of Electronics and Communication Engineering, Mepco Schlenk Engineering College, Sivakasi, Tamil Nadu, India

*Address all correspondence to: senthilkumarj@mepcoeng.ac.in

IntechOpen

© 2019 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/3.0>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. 

References

- [1] Huang S, Tan KK. Hardware-in-the-loop simulation for the development of an experimental linear drive. *IEEE Transactions on Industrial Electronics*. 2010;57(4):1167-1174. DOI: 10.1109/TIE.2009.2038408
- [2] Martin A, Emami MR. Dynamic load emulation in hardware-in-the-loop simulation of robot manipulators. *IEEE Transactions on Industrial Electronics*. 2011;58(7):2980-2987. DOI: 10.1109/TIE.2010.2072890
- [3] Pouria S, Samereh Y. State of the art: Hardware in the loop modelling and simulation with its applications in design, development and implementation of system and control software. *International Journal of Dynamics and Control*. 2015;3(4):470-479. DOI: 10.1007/s40435-014-0108-3
- [4] Robin C, Emami MR. A holistic concurrent design approach to robotics using hardware-in-the-loop simulation. *Mechatronics*. 2013;23(3):335-345. DOI: 10.1016/j.mechatronics.2013.01.010
- [5] Senthil Kumar J. Investigation of adaptive controllers for precise trajectory tracking of three link planar rigid robotic manipulator employing hardware-in-the-loop simulation [thesis]. Chennai: Anna University; 2017
- [6] Maass J, Kohn N, Hesselbach J. Open modular robot control architecture for assembly using the task frame formalism. *International Journal of Advanced Robotic Systems*. 2006;3(1):1-10. DOI: 10.5772/5763
- [7] Tegin J, Wikander J. Tactile sensing in intelligent robotic manipulation—A review. *Industrial Robot*. 2005;32(1):64-70. DOI: 10.1108/01439910510573318
- [8] Baeten J, Bruyninckx H, De Schutter J. Integrated vision/force robotic servoing in the task frame formalism. *International Journal of Robotics Research*. 2003;22(10-11):941-954. DOI: 10.1177/027836490302210010
- [9] Bicchi A, Kumar V. Robotic grasping and contact: A review. In: *IEEE International Conference on Robotics and Automation*. San Francisco, CA; 2000. pp. 348-353
- [10] Borst C, Ott C, Wimbock T, Brunner B, Zacharias F, Bauml B, et al. A humanoid upper body system for two handed manipulation. In: *IEEE International Conference on Robotics and Automation*. 2007. pp. 2766-2767
- [11] Gunji D, Mizoguchi Y, Teshigawara S, Ming A, Namiki A, Ishikawaand M, et al. Grasping force control of multi-fingered robot hand based on slip detection using tactile sensor. In: *IEEE International Conference on Robotics and Automation*. Pasadena, CA, USA; 2008. pp. 2605-2610
- [12] Han L, Trinkle JC, Li Z. Grasp analysis as linear matrix inequality problems. *IEEE Transactions on Robotics and Automation*. 2000;16:1261-1268. DOI: 10.1109/70.897778
- [13] Enea S, Markus K, Tinne DL, Herman B, Marcello B. Intelligent Robots and Systems (IROS) Preview coordination: An enhanced execution model for online scheduling of mobile manipulation tasks. In: *IEEE/RSJ International Conference*. 2013. pp. 5779-5786
- [14] Timothy J. Mobile manipulation for the KUKA youBot platform [thesis]. Worcester Polytechnic Institute; 2013
- [15] Di Napoli G, Filippeschi A, Tanzini M, Avizzano CA. A novel control strategy for youBot arm. In: *IECON—42nd Annual Conference of the IEEE Industrial Electronics Society*. Florence, Italy; 2016. pp. 482-487