Bilkent University
Department of Computer Engineering

Senior Design Project

LibreBot

Project Specifications Report

Team Members

21901779 Muhammed Can Küçükaslan

21803473 Mustafa Utku Aydoğdu

21802880 Muhammet Hikmet Şimşir

21802530 Mustafa Yasir Altunhan

21902298 Giray Akyol

Supervisor: Salih Özgür Öğüz

This report is submitted to the Department of Computer Engineering of Bilkent University in partial fulfillment of the requirements of the Senior Design Project course CS491/2.

Table of Contents

1. Introduction

1.1 Description

Our project's purpose is to model a robot that automates the process of placing a delivered book on the shelves, finding and bringing the desired book to the user, and maintaining the book order of the library. When a book is returned the robot will be able to place the book in the correct shelf in the library. The robot also can collect unattended library books from the study desks and put them on the appropriate shelves. Besides those, it will be able to bring the desired book that was specified by its "call number" from a user. To specify a call number to the robot, the user will log in his/her library account, assuming the library management integrated control instructions of the robot to their library website with our help, and add its desired book to the queue of books to fetch. The robot will also scan through the shelves to check whether there are any books placed on wrong shelves or in wrong order regularly -with time interval specified by the library personnel. The library management and a normal user will have different privileges for the control of the robot. For instance, the library manager may command the robot to look for the shelves and adjust the order of the books if their order is not correct whereas a normal user may only be authorized -indirectly– to issue a book fetch request in the system.

Normally, building such a robot involves heavy work for both hardware and software components. However, our project is focused on the software part of the robot that involves but is not limited to computer vision, object detection, object recognition, decision-making for both low and high-level decisions, motion planning under specific constraints (either using solvers for linear and non-linear optimization problems or using reinforcement learning). We will use the drake robotics framework [1] to design the robot model, to simulate the behavior of the robot, to simulate the 3-D environment involving robots, bookshelves, books, desks, people, and other 3-D objects, and to test the functionalities.

One of the innovation types that our project is related to is service. After our project is finished and implemented in a library system, a user of a library will no longer need to look for the exact location of a book in the library system, walk into the chamber shelf where the desired book is located, take the book and register the book into the library system but simply issue a fetch request of a specific book in the library system. Therefore, it will ease the job of a user.

From time to time, users of the library may put a book on the wrong shelf or position so the library personnel may need to walk around and check the shelves to check if there are any books incorrectly located and put it into the correct position. However, with the help of the robot, this task of the library personnel may be delegated to the robot which makes the life of the library personnel much easier.

The scope of the change of our project is transformation rather than optimization because its main aim is not to improve the productivity of an already existing mechanism of fetching and placing library books but to propose a totally new solution that take advantage of the state-of-art autonomous robot technologies to place returned books into the shelves, to fetch desired books from the shelves and to keep the correct order of the books in the shelves.

To illustrate how the robot works, consider a case where the robot places a book left on a table into the correct position of the library. First, the robot moves around the library to find any book left on a desk. When the robot encounters a desk with no one using it for at least 45 minutes (the amount of time can change according to a specific library rule), it stops next to the desk and runs an object detection algorithm for the items on the desk. The reason why the robot runs an object detection algorithm is that it should not pick a random object such as a pencil or water cup, but pick a book. After the execution of the object detection algorithm, it should pick any one of the books left using a motion planning algorithm to be able to grasp the object successfully without harming the environment. For instance, there may be people nearby so the robot should have a restricted working area and restricted angles for the movements of its arms in order not to hit any person or object nearby. Besides that, the book may not easily be grasped since there is no space between the book and the desk so the robot may move the book to a corner of the desk and grasp from there which requires different types of 6 or fewer degrees of freedom [2] in different parts of the motion. After that, an object recognition algorithm is run to check whether it is a user book or a library book. The plan, for now, is that the internal structure of the object recognition algorithm will benefit from the area of computer vision and look for a “call number” written on the book which is used to identify library books. After that, the robot needs to go to the correct position of the book in the library, which may or may not require the robot to go to a different floor or chamber. For the case of the robot going to a different flat, decision-making is used to decide which path to take in order to go to the specific flat and chamber. Feedback control is necessary to make sure the robot goes to the right place without harming or restricting its environment and itself. Each step or movement may give feedback to the robot so that it takes a better-optimized step or motion in consecutive movements. After finding and going to the right flat and chamber, the robot places the book into the correct position again using decision-making, motion planning, and object recognition (to find the exact spot of the correct position of the book by looking at the nearby books).

1.2 Constraints

1.2.1 Implementation Constraints

- LibreBot software and ML algorithms will be written using python programming language with the help of the drake robotics framework.
- For perception and TAMP, ML algorithm libraries in Python programming language will be used.
- pydrake bindings of Drake framework will be used for the simulation of the robot and the environment. Drake framework's optimizer and symbolic solvers will be used for control and manipulation.
- The collaboration will be done using Jupyter notebooks from providers such as Google Colab or Deepnote.
- Code version control will be done via git with GitHub as the git storage provider.
- The 3D model of the robot and the properties that exist in the real world will be stored as URDF or SDF files.

1.2.2 Sustainability Constraints

- If there are changes in the robot's operating environment such as increasing or decreasing bookshelf spacing, bookshelf height, shelf width, shelf depth, number of shelves per bookshelf, etc., then the simulation environment may need to be updated to reflect the real-world changes. Following the simulation environment updates, TAMP and perception algorithms may need to be redesigned/retrained as well.

- Although it is a low-probability event, the drake framework may get deprecated or abandoned; in that case, the simulation and optimization/solver library need to be switched to an alternative framework.
- Periodic maintenance of the robot will be required to prevent damage/loss of equipment and the safety of humans.

1.2.3 Time Constraints

- In the first month of the project (by November), the robot's control mechanism on the Drake framework will be implemented.
- In the second month of the project, distance calculation from camera image and book insertion will be implemented.
- In the last month of the project, object detection using the image data from the camera will be implemented.
- The first demo will be demonstrated in mid-December 2022.
- We estimate that the project will be completed by May 2023.
- To complete the project we plan on learning and getting experience using the drake framework. To do this, the Robotic Manipulation course by Russ Tedrake [3] will be followed to get an understanding of concepts such as control, pick & place, pose estimation, etc. Online lectures will be followed, and problem sets and exercises of the course will be solved.

1.2.4 Social Constraints

- Libraries are quiet spaces and to fit in the environment there should be limits on the noise the robot makes. To not disturb people, distracting elements such as lights, excessive movement, and vibrations should be minimized.
- Some predictions estimate that 38% of US jobs could be automated within 10 years [4]. Many people are concerned due to fear of losing their jobs and call against automation [4]. This fear may also affect the adaptation of the LibreBot to libraries. We should clearly explain and convince the public that the automation that LibreBot will improve the productivity of the librarians and it will also create new jobs. We should also convince the public that the number of created jobs will be enough to cover the number of lost jobs [5].

1.2.5 Language Constraints

- The interfaces and documentation of the robot will be in English.

1.2.6 Technological and Manufacturability Constraints

- LibreBot needs several ML algorithms that need to be trained, depending on the computational power of the hardware, the algorithms may not be trained to their highest potential. As the availability of computational power increases, the algorithms may be further optimized.
- To operate the robot needs a source of energy. This may be electrical energy provided from wall outlets and connected via an electric cable or internal power storage such as batteries. The amount of power limits the capabilities of the robot as it limits both the computational power of the hardware and the physical size and strength of the manipulation arms. LibreBot may use a hybrid combination of the two energy sources.
- LibreBot should have a weight limit to not damage the floor which consequently limits the weights of devices mounted on the robot. As a result, LibreBot may not be able to carry optimal hardware and instruments. To satisfy the limits the robot may need to be mounted with subpar equipment which may hamper the robot's perception and/or movement abilities.
- The robot should have a size limitation as well. As libraries are optimized for humans a robot too big may have difficulty navigating around bookshelves and tables. To keep the robot's size in check the quality and competence of the devices installed on it may suffer.

1.2.8 Privacy & Security Constraints

- The camera of the robot may occasionally show the humans. So it should either not record at all, or delete the parts of the recordings that may show humans.
- The robot camera should be secure enough to not let third-party applications control the camera and record the video.

1.2.9 Environmental Constraints

- In order to train the ML algorithms considerable computational power is needed which uses a significant amount of energy in the form of electricity. This energy may come from fossil or non-renewable sources that impact the environment negatively. Inefficient training algorithms and hardware should not be preferred.

- As the robot itself uses electricity, energy efficient motors and lightweight materials should be preferred. Use of heavy metals that damage the environment should be minimized, recyclable materials should be used instead.

1.2.10 Economic Constraints

- As the budgets of libraries are limited the robot should not be made of unnecessarily expensive components and a balance must be struck between features and price.
- One of the most important components of the system is the ML algorithms and to train these algorithms substantial computational power is needed, the hardware to train the algorithms is expensive and may limit the capabilities of the robot.

1.3 Professional and Ethical Issues

- Simulations will be performed in the Drake toolbox.
- The source code will be accessible privately to the group members and graders in Deepnote and GitHub. The private source code should not be shared with third parties until the end of the project.
- The software frameworks and libraries should be used and given credit in compliance with the license requirement.
- Every Friday, preferably face-to-face group meetings will be scheduled with the team and the supervisor.
- Each member of the group is expected to attend weekly meetings.
- Each member of the group will take part in the project development lifecycle and decision-making process of the group.
- Each group member should research the topics related to his work. Group members are supposed to help each other and get help from the supervisor if they need it.
- The work division should consider the members' interests and should be made as just as possible.

1.4 Risks

- LibreBot may have difficulties picking up and manipulating books that are open or significantly damaged (e.g., torn cover, torn pages, wet book, sticky book). This is because it is not feasible to simulate a training environment in which the physics of individual pages of a book are calculated.
- LibreBot may encounter an unlabeled book in its operation, either on a shelf or on the ground, in this case, the robot should notify personnel of the approximate location of the unlabeled book.
- There is the problem of book alignment, the book on shelves may slide into a horizontal or diagonal position. This poses a problem when LibreBot is picking, placing and identifying books. In such a situation the robot may get stuck and unable to pick up a requested book as a solution LibreBot may invalidate the request and/or notify personnel.
- LibreBot may drop a book while it is picking it up or transporting it. In case this happens there is a chance that the robot may not be able to pick the book back up. In such a case the robot may ask for assistance or communicate to the personnel that the book is dropped and continue operating without picking it back up.
- While operating, there is a chance the robot falls over. This may be caused by insufficient planning by the robot or by other factors such as human/furniture hitting the robot. To not put humans at risk or damage themselves, the robot should cease movement, notify personnel, and shut down if necessary.
- LibreBot has to navigate around and operate in the tight spaces and sharp corners of the library environment. While navigating in the library or picking up and manipulating objects there may be cases in which the robot hits an object in its surroundings. As a result, the LibreBot and/or the object hit may be damaged. In the event this happens, the robot should transmit information about the accident (e.g., forces exerted, mechanical arm speed, etc.) to the personnel. It is also possible to limit the maximum force and speed the arm can move.
- Libraries are used by people so there will be instances of the robot coming close to a human, in such an instance the robot may put the human at risk of being hit by the arm or driving into the human. To prevent such risks LibreBot may cease operation when a person gets close to the robot, and hard limits may be put on the maximum speed the

robot can operate at to mitigate the risks further. LibreBot may only be permitted to run when the library section is closed to humans (e.g., when the library is closed, or rotating closures of library sections).

- The robot is not able to ascend/descend stairs, ladders, and similar obstacles. LibreBot may not be able to deliver books from one floor to another, it is best operated only in the same library room.
- It might be difficult to train some of the machine learning models under time and budget limitations.

1.5 Alternatives in the Market

AuRoSS is a system proposed by Li, Huang, et al. [6] that strives to solve the library book tracking problem LibreBot also aims to solve. A significant difference between LibreBot and AuRoSS is how the systems identify books. While both LibreBot and AuRoSS use RFID tags, LibreBot will also use computer vision and text recognition when RFID is not available. If the book does not have RFID tags LibreBot requires that the book label is not damaged, not obstructed and the environment has sufficient lighting to extract the information. On the other hand, RFID tags require high-accuracy positioning of the scanner and enough range of the scanner [6]. In general, libraries already have labels on books but RFID tags are new technology (by library standards). While scanning RFID tags may be quicker, to use AuRoSS, libraries must tag the books, which costs time and money. Compared to AuRoSS, LibreBot is an end-to-end system, it handles the book-handling process from an electronic book request to its delivery. AuRoSS only handles missing and misplaced books while LibreBot handles gathering the books from shelves, placing back borrowed books, and correctly sorting misplaced books. LibreBot's scope is much larger and it is a more complete system compared to AuRoSS.

2. Requirements

2.1 Functional Requirements

2.1.1 System Functionalities

System functionalities refer to the library robot's capabilities developed for conducting specific tasks inside the library. The functionalities of the library robot consist of several core parts.

2.1.1.1 Perception

The library bot makes decisions according to the environment around it. Hence one of its core functionalities is the perception of its environment. The library robot should be able to:

- Perceive books with different shapes (e.g. thin, large)
- Perceive desks in the library from which it can collect the books
- Perceive bookshelves in the library to collect books from or place books back on to
- Perceive obstacles such as humans, chairs, and trash boxes to avoid any collision with these
- Recognize irrelevant objects on the desks, such as notebooks, computers, books that do not belong to the library, and other personal accessories of the students. This ability will prevent robots from taking away incorrect types of items from desks to bookshelves.
- Detect whether a book has a call number on it. The call number is the text or code used to identify books in the library classification systems. If this text is detected, the robot will attempt to take the book from the desk and put it on the library shelves; otherwise, it will not make such an attempt.
- Recognize the call number to identify books belonging to the library. This recognition of the unique text will enable the robot to perform many functionalities, such as distinguishing different books and understanding what part of the library a book belongs to.
- Recognize the texts written on the library shelves to find the correct shelve to insert (or take from) the book.

- Perceive the overall structure (e.g., whether books are placed horizontally or vertically) of a particular environment (e.g., desk or bookshelves) to make decisions regarding the actions that will be taken to place books.

2.1.1.2 Decision-Making and Interaction

The library robot work will continue on the decision-making and interaction stage after perceiving the environment around it. The library robot should be able to:

- Take the necessary command from the users (via integrated library system), as input to the LibreBot software system, to conduct the following:
  - Bring the specific book, which is specified as input by the user, from the library shelves
  - Put back a specific book, when the books are returned, from the returned book desk to the shelves
- Move from one location to another within the library to conduct necessary tasks.
- Stop the motion when an obstacle is observed along the path, change the path if needed, and continue to the motion when the obstacle is no longer in the path.
- According to the perception data regarding the positioning of the book, make the necessary move to grasp the book. This move might consist of several sequences such as first altering the current position of the book to convert it into a more desirable position for seizing it, followed by the actual seizing act.
- According to the perception data regarding the overall structure of the library shelves, make the necessary moves to insert the book. This move might consist of several sequences, such as first creating a free space between other books to insert the book, and then actually inserting the book.
- According to the text data written on books (unique “Call Number” in case of Bilkent University) or bookshelves, stop in the shelf that the book needs to be placed while moving through the bookshelves.
- Aligning the books that are not aligned properly in the library shelves according to the positioning data. If the robot encounters an incorrect ordering of the books when aligning them, it removes the book breaking the order of appropriate library classification system and inserts that book into a correct position.

2.1.2 User Functionalities

The LibreBot system will be developed as a software system. Hence, users' interaction with a robot (i.e commands and requests given to the robot) will be provided through the input sent by the user to the LibreBot system.

Users of the system interacting with the library robot will be able to:

- Request a book from the library system integrated into LibreBot's system. Then the call number and other relevant details will be conveyed to the robot by the system. Finally, the robot will bring the book using this information.
- Command the robot to put returned books back on the library shelves over the system. The robot will identify and pick the book, then request the relevant information, such as the shelf location, from the library system. Finally, it will carry the book back to its original place.
- Command the robot to align the books on the shelves. When this command is given, the robot will align the books on the shelves in the library (e.g a book standing in an inclined position, the robot will make the book aligned with the bookshelves again). The robot also should be aware of the normal differences due to the size of the books. If the robot encounters an incorrect ordering of the books when aligning them, it removes the book breaking the order of appropriate library classification system and insert that book into a correct position.
- For the safety of the people in the vicinity of the robot, the robot will have emergency halt buttons.

2.2 Nonfunctional Requirements

2.2.1 Performance and Efficiency

The library robot system should:

- Operate in real time to conduct interaction between the environment and itself. Therefore the robot is required to conduct perception and recognition tasks, then make the necessary decisions and act accordingly in real time.

2.2.2 Usability

- The library robot should be easy to communicate with, and the model should be integrated into the library's existing digital system.

2.2.3 Extensibility and Portability

- The library robot model should be able to adapt different shelving arrangements in libraries
- The library robot model should be able to adapt to different library classification systems.

3. References

[1] Russ Tedrake and the Drake Development Team, “Drake: Model-Based Design And Verification For Robotics,” 2019. [Online]. Available: https://drake.mit.edu. [Accessed: 16-Oct-2022].

[2] Wikipedia, “Underactuation,” Wikipedia, 17-Oct-2022. [Online]. Available: https://en.wikipedia.org/wiki/Underactuation. [Accessed: 16-Oct-2022].

[3] R. Tedrake, “Robotic Manipulation: Perception, Planning, and Control.” [Online]. Available: http://manipulation.mit.edu. [Accessed: 16-Oct-2022].

[4] A. Quart, “Automation Is A Real Threat. How Can We Slow Down The March Of The Cyborgs?,” The Guardian, 08-Aug-2017. [Online]. Available: https://www.theguardian.com/us-news/2017/aug/08/humans-v-robots-defending-jobs. [Accessed: 16-Oct-2022].

[5] I. Wladawsky-BergerGuest, “Automation And The 2030 Job Hunt,” WSJ, 09-Feb-2018. [Online]. Available: https://www.wsj.com/articles/automation-and-the-2030-job-hunt-1518198504?tesla=y. [Accessed: 16-Oct-2022].

[6] R. Li, Z. Huang, E. Kurniawan, and C. K. Ho, “AuRoSS: An Autonomous Robotic Shelf Scanning system,” in 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Sep. 2015, pp. 6100–6105. doi: 10.1109/IROS.2015.7354246.