Autonomous Robot for Inventory Management in Libraries

Rushad Mehta
Department of Mechatronics Engineering
SRM Institute of Science and Technology
Chennai, India
rushadmehta16@gmail.com

Abhay Sahu
Department of Electronics and Communication Engineering
SRM Institute of Science and Technology
Chennai, India
abhay.sahu10@gmail.com

2020 IEEE International Students' Conference on Electrical,Electronics and Computer Science (SCEECS) 978-1-7281-4862-5/20/$31.00 ©2020 IEEE 10.1109/SCEECS48394.2020.18

***Abstract*—This paper proposes an autonomous robot to reduce labor, increase the efficiency of existing inventory management systems and provide a solution to develop an automated inventory. Managing a college library comes under inventory management. Like an inventory management system, the library also faces problems with the availability of books. For example, when a book gets returned, the database gets updated and indicates that the book is available, but one cannot find the book in its location as it has not been placed there. The time and energy of the librarian are wasted as the books have to be placed back to its location. These problems are solved by designing an inventory management robot that is implemented in a library to put the returned books back to its location. The returned books are placed on the shelves of the robot. The robot scans the Radio Frequency Identification (RFID) tag of books to get the shelf location of the books and then navigates through the library to the location where the book should be and places the book on its shelf thus making the book available quickly. The innovation of this system is that since it works in a known environment, it does not run on complex SLAM techniques, thus reducing the complexity of the system while being relatively flexible. This implementation also does not require any significant changes to be made to its environment.**

***Index Terms* - autonomous robot, inventory management, library robot, path planning, autonomous navigation**

I. INTRODUCTION

Many problems come to mind when we think about inventory management, such as availability monitoring, speed of storage and retrieval of goods and automation. A college library is a type of inventory and often has a similar homogeneous design like warehouses. Problems such as a delay in updating the library website after a book is returned or being unable to find a book in its location as it is still at the librarian's desk, similar to what happens at a warehouse when goods come to store or when someone demands goods placed inside a warehouse.

Several techniques have been proposed in the past to overcome these challenges. The simplest solution to this problem is a line following robot [1]. The major drawback of this system is that it needs lines to be drawn on the library floor, reducing flexibility. The next technique overcomes the problems associated with line-following robots by employing navigation algorithms [2]. Such algorithms do not find the optimal path between the source and destination, reducing the efficiency of the system. Alternative approaches only check for the position of books for their easy localization [3]. Some approaches use expensive control boards such as myRIO and are not autonomous [4]. To achieve the required management of the library, the approach needs to be autonomous. These robots are autonomous and are also capable of shelving books at different heights of a bookshelf, but use a camera, increasing cost and complexity [5]. All of the above techniques require changes to the existing library environment for implementation. Existing library robots do not solve all of these problems efficiently. All these robots are designed to be used in libraries only and cannot be used in warehouses for inventory management.

This paper proposes an alternative method that is capable of doing all of the above and can travel efficiently covering the shortest path even if it has to shelve multiple books at a time and can also be used in warehouses. A library has straight parallel paths. One can imagine it as side by side parallelograms, as angles need not be at right angles. The robot uses a predefined map designed to travel across the library. The robot is designed in a way that it is capable of storing three books at three different bookshelves at a time. The robot is equipped with a platform to place books in it whose height is adjustable using a scissor elevator mechanism. Additionally, the robot has an RFID scanner to scan the books RFID tags, a WiFi module to send and receive sensory data and commands from the data processing unit. Since the heavy processing is not done on board to save battery power, high torque motors with rotary encoders, an Inertial Measurement Unit (IMU), a magnetometer and range measurement sensors are on the robot. To eliminate noise, the moving average filter is used at the IMU's output. A complementary filter is used to get the best fit output by combining the rotary encoder and IMU's output to get a precise measure of the distance traveled by the robot. Dijkstra's algorithm is used to decide the shortest path for placing all books in one round.

After the complete setup of the robot in the library, it is fully autonomous and takes books as an input.

Fig. 1 shows the flowchart of the full process.

978-1-7281-4862-5/20/$31.00 ©2020 IEEE

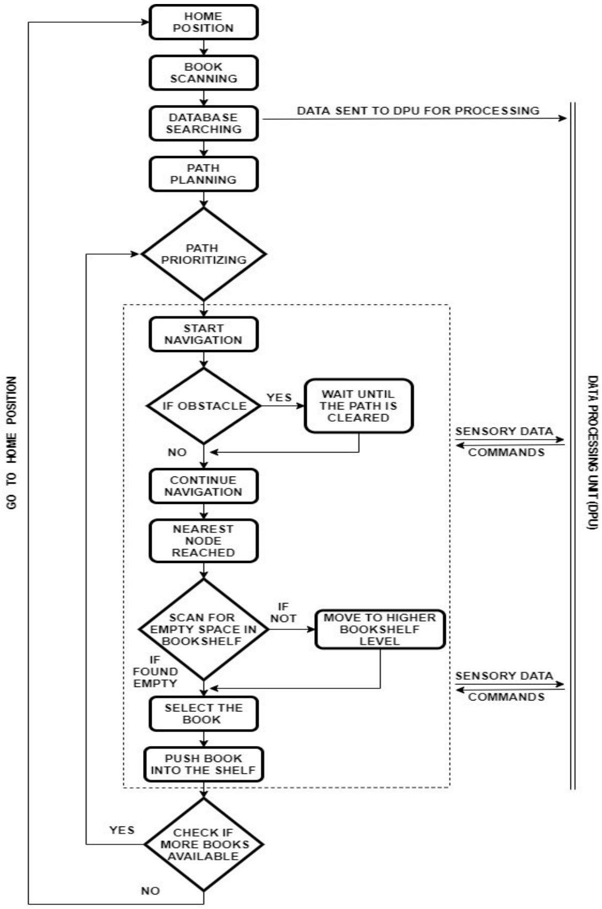

Fig. 1. Process Flowchart

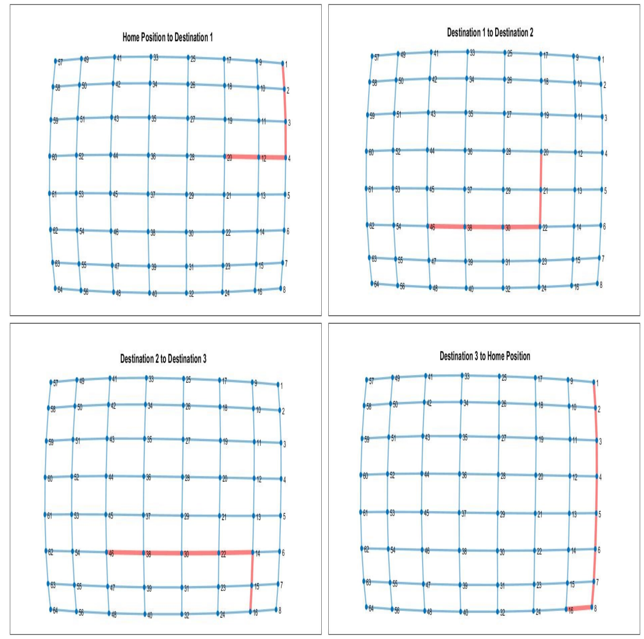

Fig. 2. Path planning for three books

II. PATH PLANNING

For the robot to move autonomously in its environment, it should be able to localize itself and find the best-suited route from its current location to its goal location. For accomplishing this task, Dijkstra's algorithm [6], [7] is used to find the shortest path between any two given nodes. An algorithm creates two adjacency lists that represents the map of the known environment ie. the library. It creates a square graph as shown in Fig. 2 of size $m \times n$, where n is the number of rows and m is the number of columns. Additionally, the nodes have Cartesian coordinates associated with their node numbers. For eg., node 1 will have coordinates (0,0) and node 2 will have coordinates (1,0). The i^{th} node of the graph will have coordinates ($i\%n - 1, round(i/n) - 1$), where % is the modulo operator. For the robot to place x number of books in one round, it firstly calculates all the individual paths and their weights and then travels to the node associated with the least weighted path from its home position. The position of the robot is updated and the next least weighted path is found. This process continues until it places all the books and returns to its home position.

Algorithm 1 Dijkstra's Algorithm

function Dijkstra(*Graph*, *source*):
dist[*source*] := 0
for all vertex v in *Graph* do
  if $v \neq$ source **then**
    dist[v] := *infinity*
  end if
  add v to Q
end for
while Q is not empty **do**
  v := vertex in Q with min dist[v]
remove v from Q
  for all each neighbor u of v **do**
    alt := dist[v] + length(v, u)
    if *alt* <dist[u] **then**
      dist[u] := *alt*
    end if
  end for
end while
return dist [ ]
end function

III. ROBOT KINEMATICS AND CONTROL

The inverse kinematics solution of the robot as shown in Fig. 3 is described as [8]- [11]:

$$\begin{pmatrix} \phi_r \\ \phi_l \end{pmatrix} = \begin{pmatrix} \frac{1}{r} & 0 & \frac{b}{r} \\ \frac{1}{r} & 0 & \frac{-b}{r} \end{pmatrix} \times \begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} \tag{1}$$

Where φ_r and φ_l are the angular velocities of the right and left wheels respectively, r is radius of the wheel, b is the distance from the wheels to the robots' frame of reference, $\dot{x}$ and $\dot{y}$ are the velocity components along x and y, and $\dot{\theta}$ is its angular velocity.

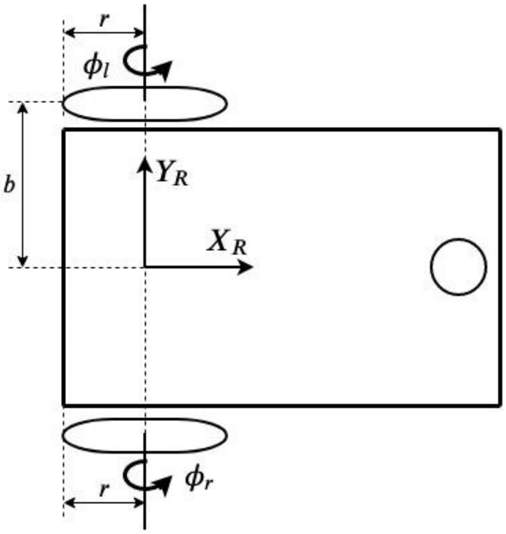

Fig. 3. Wheel Kinematics

To ensure that the robot navigates without jerking, a variation of the Gaussian function [12]- [14] is employed. This ensures that the velocity profile of the robot is smooth and would not suffer from jerks.

$$f(x_{current}|x_{total},\, \sigma^2) \;=\; \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-\left(x_{current}-\frac{x_{total}}{2}\right)^2}{2\sigma^2}} \qquad (2)$$

Where $x_{current}$ is the current distance travelled by the robot in a straight path, x_{total} is total distance to be travelled in straight path and σ is an arbitrary constant that defines the rate of change of the robots' velocity. Fig. 4 shows the velocity profile of the robot.

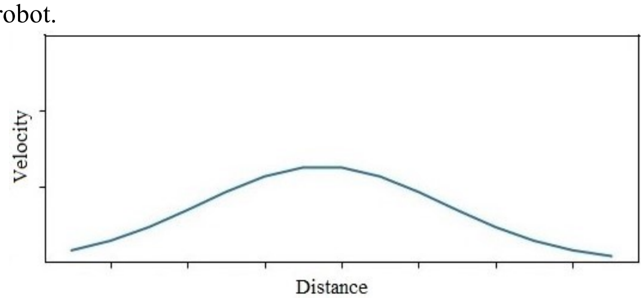

Fig. 4. Velocity profile of the robot

IV. AUTONOMOUS NAVIGATION

For the robot to localize itself, it needs to accurately measure the distance and direction it travels in. To increase the accuracy of the sensors [15], the complementary filter and the moving average filter [16] are used. The IMU provides us with the acceleration and angular velocity, which is double integrated to calculate the distance traveled and the heading direction of the robot. The rotary encoders [17] provide the distance traveled by each motor. Since the accelerometer readings are very susceptible to noise, the readings are firstly fed to a moving average filter, which reduces the noise and smoothens the data. The resulting data of the moving average filter and encoder is fed to a complementary filter which provides the best-fit estimate of the distance traveled.

The moving average filter is mathematically described as:

$$y_s(i) = \frac{1}{2N+1}\,[\,y(i+N) \;+\; y(i+N-1) \;+\cdots+\; y(i-N)\,] \quad (3)$$

where $y_s(i)$ is the smoothened value for the i th data point, N is the number of neighboring data points on either side of $y_s(i)$, and $2N + 1$ is the span.

The complementary filter is mathematically described as:

$$y = \alpha \times y + (1 - \alpha) \times x \qquad (4)$$

where α is the filter parameter (chosen to be 0.90) and x and y are the quantities to be filtered, thus reducing (4) to:

$$dist = 0.90 \times dist_{encoder} + 0.1 \times dist_{accelerometer} \qquad (5)$$

Fig. 5 is the generalized representation of a library's layout. For the robot to estimate it's orientation, it uses a gyroscope and a magnetometer [18]. (6) estimates the amount by which the robot has to rotate to align itself along an axis. Where α is the angle between axis 1 and zero degrees of magnetometer and θ is the angle between axis 1 and axis 2 or axis 3 and axis 4.

$$\begin{aligned} headingDirection_{axis1} &= \alpha \\ headingDirection_{axis2} &= \alpha - \theta \\ headingDirection_{axis3} &= \alpha + 180 \\ headingDirection_{axis4} &= \alpha - \theta + 180 \end{aligned} \qquad (6)$$

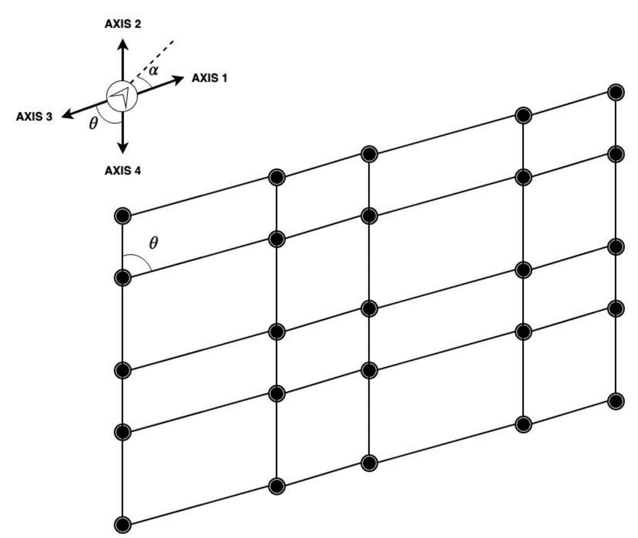

Fig. 5. Defining Axes

Algorithm 2 Autonomous Navigation Pseudocode

```
headingDirection = axis1
route = Dijkstra(Graph, origin, destination)
for all node in route do
    coordinate_x = (route[node] % n) – 1
    coordinate_y = round(route[node] / n) – 1
end for
if next(coordinate_x) > current(coordinate_x) then
    if headingDirection == axis1 then
        moveForward()
    else
        headingDirection = axis1
        moveForward()
    end if
else if next(coordinate_x) < current(coordinate_x) then
    if headingDirection == axis3 then
        moveForward()
    else
        headingDirection = axis3
        moveForward()
    end if
else if next(coordinate_y) > current(coordinate_y) then
    if headingDirection == axis2 then
        moveForward()
    else
        headingDirection = axis2
        moveForward()
    end if
else if next(coordinate_y) < current(coordinate_y) then
    if headingDirection == axis4 then
        moveForward()
    else
        headingDirection = axis4
        moveForward()
    end if
end if
```

V. STORAGE MECHANISM

The storage structure includes a scissor elevator with a platform to place books side by side, a pushing mechanism, an RFID [19], [20], [21] reader and a range measurement sensor. The pushing mechanism consists of two threaded rod motors, one of which is used to push the book into the bookshelf and the other to move the pushing motor to select the desired book. The RFID reader is placed such that the book's RFID tag gets scanned while placing the books on the platform. The autonomous storage system starts when the robot reaches the node nearest to the destination bookshelf and turns towards the bookshelf. First, it detects the starting of the bookshelf using the range measurement sensor. As the bookshelf is detected the robot moves forward while scanning the bookshelf for spaces. If the shelf is empty, the desired book is selected using the threaded rod motor and pushed into the bookshelf. If there is no space on the first level of the bookshelf the scissor elevator mechanism increases the height of the platform to the next level and the process repeats until space is found on the shelf. The distance covered along the shelf before detecting the space is monitored and used to get to the next nearest node. After reaching the node the robot navigates to the next book location. This process repeats for each book.

VI. DATA PROCESSING UNIT (DPU)

The Data Processing Unit (DPU) manages where the sensory and navigation processing is done. Algorithms need high processing power and if these computations are performed onboard, the onboard processor will consume high power. To overcome this power demand, the processing is not done on the robot. Instead, it is done at the charging station having a continuous power supply. A powerful processor that is in continuous communication with the robot over a network is used. The sensory data of the encoders, IMU, and range measurement sensors are collected by the onboard microcontroller and transmitted to the DPU. This data is processed in the DPU and generates appropriate commands, which are then transmitted to the robot.

VII. EXPERIMENTAL RESULTS

A. Real-time filtering process

Fig. 6 shows the real-time encoder and IMU data along with the complementary filter data.

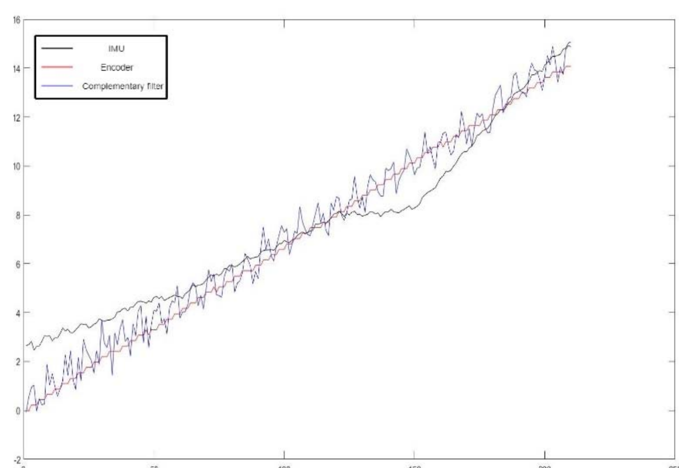

Fig. 6. Complementary filter after applying moving average filter

B. Bookshelf scanning

Fig. 7 shows the distance sensed by the robot during the scanning process for the shelf shown in Fig. 8.

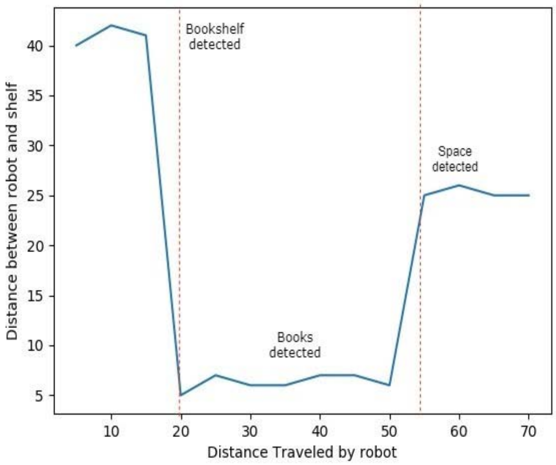

Fig. 7. Bookshelf scanning

Fig. 8. Bookshelf

VIII. CONCLUSIONS AND FUTURE WORK

Various functions of the robot are calibrated to increase the accuracy of straight motion, accurate rotations, accurate measurements of distance traveled and detection of space on a bookshelf. Now, the robot can navigate autonomously in the library when the books are placed onto its shelf. This is achieved using RFID and Dijkstra's algorithm. The robot navigation speed is optimized using the modified Gaussian function so that it navigates without jerky motion. The librarian puts books directly onto the shelf of the robot when the books are returned, the robot waits for three books or for ten minutes after the first book is placed then starts its operation. The robot has a light to indicate that it is ready for the books, or it is waiting, or a "don't touch" condition that is easy to understand. Compared to other library robots this robot does not require any change in the existing library for its implementation.

The same data processing unit can be used as a central processor if multiple robots are used for multiple floors. This robot can also be used in other places that use inventory management systems such as warehouses with little to no change required in their environment.

IX. ACKNOWLEDGEMENT

We thank SRM Institute of Science and Technology, in particular, the Department of Electronics and Communication Engineering and Aakash Research Labs, SRM IST for their continuous support and providing us with the hardware and software required to complete this research project.

REFERENCES

[1] J. Thirumurugan, M. Vinoth, G. Kartheeswaran and M. Vishwanathan, "Line following robot for library inventory management system," INTERACT-2010, Chennai, 2010, pp. 1-3.

[2] M. R. Rashid, N. Uzzaman, S. Hossain and N. K. Das Shuvra, "Development of a self-navigating algorithm for library book finder robot," 2017 3rd International Conference on Electrical Information and Communication Technology (EICT), Khulna, 2017, pp. 1-6.

[3] I. Ehrenberg, C. Floerkemeier and S. Sarma, "Inventory Management with an RFID-equipped Mobile Robot," 2007 IEEE International Conference on Automation Science and Engineering, Scottsdale, AZ, 2007, pp. 1020-1026.

[4] Y. Angal and A. Gade, "Development of library management robotic system," 2017 International Conference on Data Management, Analytics and Innovation (ICDMAI), Pune, 2017, pp.

[5] H. Yu et al., "Mobile Robot Capable of Crossing Floors for Library Management," 2019 IEEE International Conference on Mechatronics and Automation (ICMA), Tianjin, China, 2019, pp. 2540-2545. 254258.

[6] Risald, A. E. Mirino and Suyoto, "Best routes selection using Dijkstra and Floyd-Warshall algorithm," 2017 11th International Conference on Information and Communication Technology and System (ICTS), Surabaya, 2017, pp. 155-158.

[7] Mingjun Wei and Yu Meng, "Research on the optimal route choice based on improved Dijkstra," 2014 IEEE Workshop on Advanced Research and Technology in Industry Applications (WARTIA), Ottawa, ON, 2014, pp. 303-306.

[8] M G Rodd, "Introduction to robotics: Mechanics and control: John J. Craig", Automatica, vol. 23, no. 2, pp. 263-264, 1987.

[9] Siegwart, Roland, Illah Reza Nourbakhsh, and Davide Scaramuzza. Introduction to autonomous mobile robots. MIT press, 2011.

[10] Nnaji, O. Bart, "A Review of Fundamentals of Robotics Analysis and Control" R. J. Schilling Prentice Hall 1990", IIE Transactions, vol. 24, no. 1, pp. 94-94, 1992.

[11] E. Maulana, M. A. Muslim and A. Zainuri, "Inverse kinematics of a twowheeled differential drive an autonomous mobile robot," 2014 Electrical Power, Electronics, Communicatons, Control and Informatics Seminar (EECCIS), Malang, 2014, pp. 93-98.

[12] A. Gasparetto, A. Lanzutti, R. Vidoni, and V. Zanotto J. "Validation of Minimum Time-Jerk Algorithms for Trajectory Planning of Industrial Robots," Trans. of the ASME, Journal of Mechanisms and Robotics, vol. 3, No. 1, Article No. 031003, Aug., 2011.

[13] A. Y. Lee, G. Jang and Y. Choi, "Infinitely differentiable and continuous trajectory planning for mobile robot control," 2013 10th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI), Jeju, 2013, pp. 357-361.

[14] Ross, Sheldon M. Introduction to probability models. Academic press, 2014.

[15] W. Ilewicz, K. Skrzypczyk, A. Galuszka, T. Grzejszczak, D. Bereska, and M. Pacholczyk, "Estimation of Uncertainty of IMU Module Measurement Results," 2018 International Conference on Signals and Electronic Systems (ICSES), Krakow, 2018, pp. 92-95´

[16] Y. Pititeeraphab, T. Jusing, P. Chotikunnan, N. Thongpance, W. Lekdee and A. Teerasoradech, "The effect of average filter for complementary filter and Kalman filter based on measurement angle," 2016 9th

Biomedical Engineering International Conference (BMEiCON), Laung Prabang, 2016, pp. 1-4.

[17] K. Nakano, T. Takahashi, and S. Kawahito, “A CMOS rotary encoder using magnetic sensor arrays,” in IEEE Sensors Journal, vol. 5, no. 5, pp. 889-894, Oct. 2005.

[18] S. A. Ludwig and A. R. Jimenez, “Optimization of gyroscope and´ accelerometer/magnetometer portion of basic attitude and heading reference system,” 2018 IEEE International Symposium on Inertial Sensors and Systems (INERTIAL), Moltrasio, 2018, pp. 1-4.

[19] Boss, Richard W. “RFID technology for libraries.” Library technology reports 39.6 (2003): 1.

[20] J. Zhang, Y. Lyu, T. Roppel, J. Patton and C. P. Senthilkumar, “Mobile robot for retail inventory using RFID,” 2016 IEEE International Conference on Industrial Technology (ICIT), Taipei, 2016, pp. 101-106.

[21] Tao Liu and Jian Sun, “Simulative calculation and optimal design of scissor lifting mechanism,” 2009 Chinese Control and Decision Conference, Guilin, 2009, pp. 2079-2082.