# Tracking of Pallet in Automatic Storage and Retrieval system

Shobowale Kafayat Oluwatoyin* Md Raisudin Khan** Riza Muhida **

* International Islamic University/Automous Agent Research Group, Kuala Lumpur, Malaysia. Email: kshobowale@gmail.com

** International Islamic University/Automous Agent Research Group, Kuala Lumpur, Malaysia. Email: raisuddin@iiu.edu.my, muhida@iiu.edu.my

*Abstract*— **This research aims to develop an algorithm for pallet movement in the x and y direction for unit load AS/RS for parking lots. The algorithm developed will be used for executing x and y motion to store and retrieve specific pallets from a puzzle like storage system. The storage system consists of components such as Limit Switches, PIC, Gears, Geared DC motor, solenoid, shaft, rack and pinion. Limit switches are used for pallet identification during in the simulation, and the rack and pinion mechanism is used to provide a substitute to the storage and retrieval machine used in AS/RS. The controller is made up of two PIC16f877A connected to communicate through serial peripheral interface (SPI) and a graphical user interface (GUI). The modified A* algorithm is used for identifying and moving pallets in and out of the warehouse. This design can be used for other products that can be stored in a warehouse.**

*Keywords*—(E.g.) **A*, AS/RS; mechanism; pallet; parking lot; puzzle; unit load.**

## I. INTRODUCTION

Automatic storage and retrieval systems (AS/RS) are computer controlled storage systems that can automatically store and retrieve loads with high throughput [1]. Development of technology in AS/RS can be divided into four stages namely-manual, mechanical, automatic (automation) and intelligent [2]. The effective management of unit loads in AS/RS is a challenge, especially finding the locations of the unit loads. Ref. [2] worked on location identification by using a cost metricx to determine the location of unit load. Ref [3] worked on particle swarm optimization to locate unit load. This paper seeks to find an algorithm that can be used for storage and retrieval of unit load for a 3x3 layout AS/RS that makes use of a special mechanism consisting of rack-pinion and gear sets as an alternative to the storage and retrieval machine used in AS/RS. The algorithm is developed by considering the initial state of the AS/RS and a set of goal state in a state space. The A* search is a strategy for finding a shortest path from the initial state to a nearest goal, such a path is called an optimal path, it finds an optimal path using a generalization of Dijksra's algorithm [4]. Unit loads are stored and retrieved by a platform and there are 8 platforms in the warehouse. The vacant space is for moving a pallet in and out of the AS/RS. There are different classes of AS/RS but this work will focus on unit load, single command, randomized storage assignment policy, closest open location for storage, and modified A* algorithm for retrieval. The rest of the paper is arranged as follows: section II discusses related work of AS/RS as regards to unit load, single command, randomized storage, section III proposes A* algorithm for 3x3 puzzle like AS/RS configuration, section IV discusses the simulation results of 3x3 row and column of a puzzle like AS/RS using PIC simulator IDE version 6.65 and the GUI using labwindows / CVI version 9.0, and finally the paper is concluded in section V.

## II. RELATED WORKS

Automatic storage and retrieval systems (AS/RS) have been applied to various fields over the past years. The advancement in technology such as the inclusion of integrated circuits, sensors, motors, control systems, software and hardware has brought about improvement in the operation and functionality of today's AS/RS [5]. Typically, AS/RS has been applied to different functions such as in storing and retrieving unit loads; an example of this is a parking structure, storing heavy duty products, and many more. The major function of an AS/RS is in storing and retrieving. The procedure in storing is quite an easy task especially if the load can be placed anywhere in the AS/RS but the process of retrieving back the load is a labour intensive task [6] and improving the performance of retrieving can lead to large amount of savings in warehousing cost. Also the efficiency of retrieving is largely dependent on storage racks (pallets), warehouse layout and the control mechanism.

There are quite a number of studies available related to the location of unit load in an AS/RS. Ref. [1] devised an AS/RS in order to be able to implement modified selection sort algorithm. In their design each column has its independent vertical platform and each row has its own independent horizontal platform. However, their AS/RS algorithm and design was for load shuffling. They performed simulation based on flexsim to build an AS/RS. Ref [7] used particle swarm optimization to determine the optimal layout, they used their proposed AS/RS model to determine the length, width and height of the AS/RS and they used their PSO algorithm to control retrievals in the AS/RS.

References [6], [8], [9] and [10] used random open location and genetic algorithm to store and retrieve unit loads. Their performance was measured based on the total travel time and average retrieval waiting time of storage and retrieval machine. They worked primarily to reduce the time it takes for the storage and retrieval machine to store and pick unit load. Ref [11] used closest open location and immune particle swarm optimization to measure the total travel time taken for each storage and retrieval operation to be completed.

References [12] and [13] performed simulation using random open location, longest waiting retrieval and first in first out approach to measure average retrieval waiting time. Ref [14], also worked on closest open location and modified selection sort algorithm to measure response time of retrieval. Ref [15] performed simulation to track pallets using java 1.5.

Based on literature review, existing work on unit load automatic storage and retrieval system employ storage and retrieval machines to store and retrieve loads. Extra spaces are required to handle this kind of situation which increases initial cost as well as running cost of the system. Also, in the case of pallet which changes locations dynamically, it is difficult to keep track of the pallets. Therefore, this research work focuses on developing an algorithm for automatic storage and retrieval system that can handle pallets changing address dynamically.

## III. METHODOLOGY

The following assumptions are made for the AS/RS system:

- There are 9 spaces in the warehouse arranged in a 3x3 matrix out of which 8 are pallets and one is empty.
- The floor and pallets are flat and of uniform height.
- There is no other object in the warehouse, only unit load is stored.
- For storage and retrieval request, the first on the queue takes precedence.
- Pallets arrive at the input station.
- Random storage location assignment policy
- Single command
- Closest open location storage request selection rule
- Modified A* algorithm for retrieval location selection rule.

The system for testing consists of a space of dimension 30 x 15 inches and divided into 9 equal spaces which represent the parking lots. Out of the 9 spaces 8 pallets are able to move along x-y direction with 4 limit switches. The empty space (parking lot) is needed for movement (shuffling) for storage and retrieval.

<u>Assumptions and notations for the A* Algorithm</u>

r: the initial state of the AS/RS

h: a heuristic function or estimated movement cost from a given space to a final destination.

V, E: the dimension of the AS/RS

c: a positive cost function on E

g: the movement cost function from starting point to a given space on the grid following the path generated to get there.

D: the AS/RS

Goalset: a set of goal vertices in D

T: shortest path

<u>The modified A* algorithm</u>

Assuming a path from r to a goal exists

Procedure A* - search pallet (D, c, r, Goalset, h, T)

Input : D= (V, E) – the AS/RS

c, r, h, Goalset.

Output : a shortest path T rooted at r containing an Optimal path to a goal vertex.

Q( a priority queue of vertices with V having priority value f(v) = g(v) + h(v), where g(v) is the cost of shortest path p(v) from r to V currently generated. Q also contain a parent pointer from V to w Є T, where edge (w,v) belongs to p(v))

**While** Q is selected do

Dequeue vertex v in Q with minimum priority value f(v)

Add vertex v to T using parent pointer

**If** V Є Goalset **then**

**return**

**endif**

**for** all vertices w Є¢ T and adjacent to v do

if w Є¢ Q then

enqueue w with parent v and priority value f(w) = g(w) + h(w)

where g(w) = g(v) + c(v,w)

else

if f(w) >= g(v) + h(w) then

reset parent pointer of w to v and update priority value of w to f(w) = g(w) + h(w), where g(w) = g(v) + c(v,w)

end if

end if

end if

end while

return "path"

end A*- searchpallet

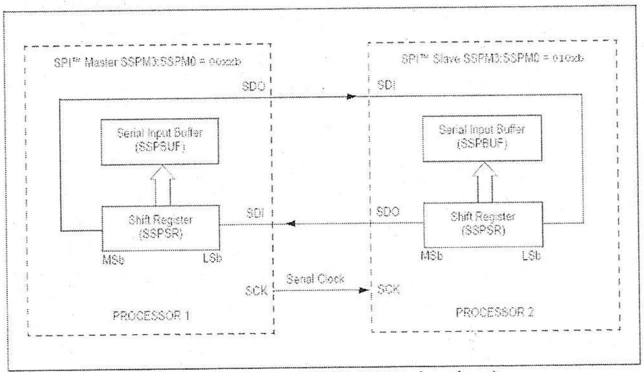

Figure 1 spi communication(source:PIC16F777 datasheet)

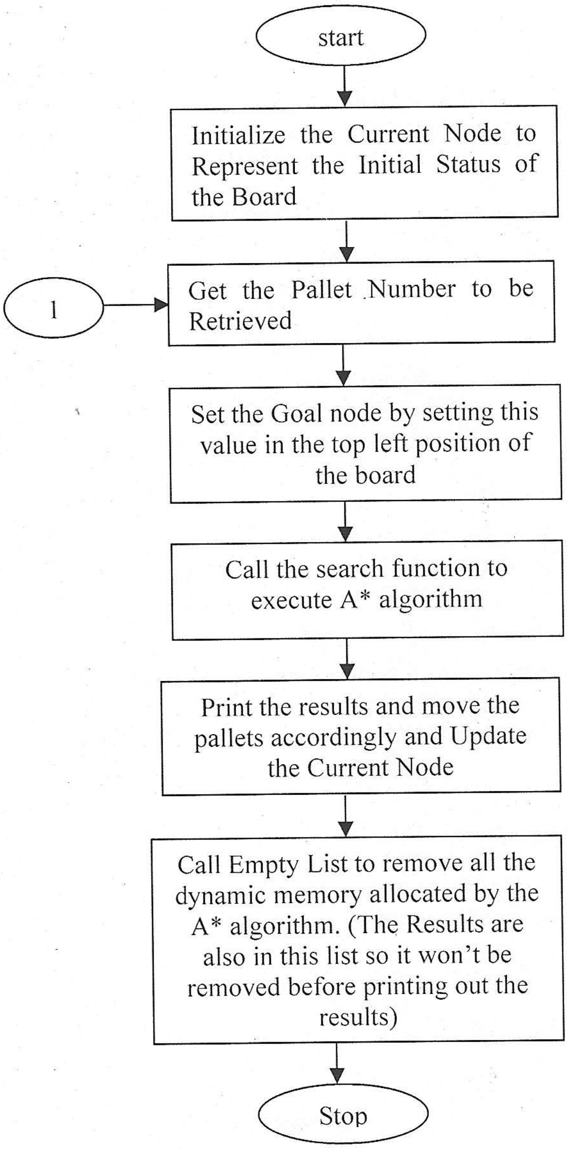

Figure 2 Flowchart for moving out a pallet from AS/RS.

The control system consists of two PIC16F877A microcontroller, one master and the other slave connected together for communication through SPI. One motor, and 15 solenoids are connected to the master microcontroller circuit while the slave is connected to the serial port to communicate with the graphical user interface in the personal computer to control the system. The GUI is developed using lab windows/CVI version 9.0.0. The layout of the AS/RS considered is as shown in the block diagram in figure 4. Each pallet has 4 limit switches for dynamic identification and for determining if the pallet is occupied or unoccupied using a binary code identity. In order to minimize the pin on the microcontroller, a 4 input OR gate has been used to connect the 4 limit switches on each pallets together. Each pallet has four caster wheels and rack-pinion mechanisms for movement along two mutually perpendicular directions (say x and y directions). When a signal is generated to move a particular pallet, if a storage or retrieval operation is to be performed, there are essentially four main operations to be performed which are:

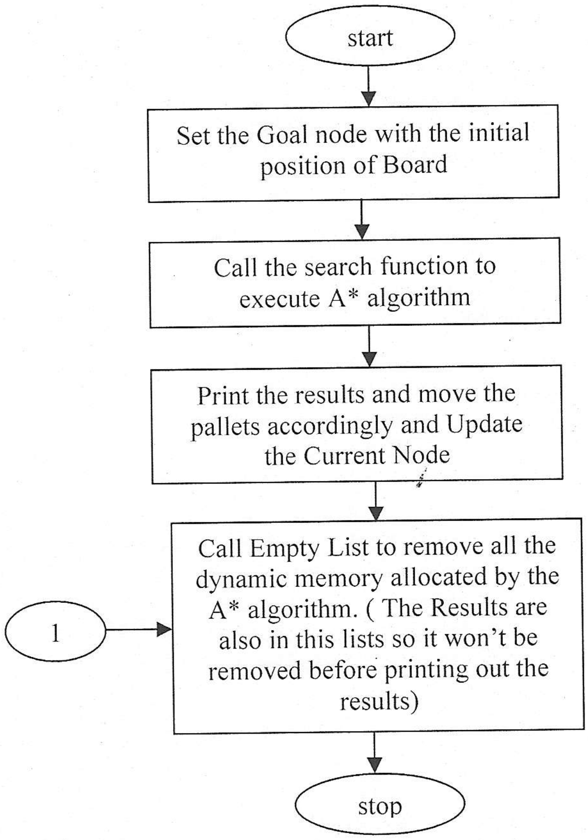

Figure 3 Flowchart for returning a pallet into the AS/RS .

-operation 1
This section deals with creating, adding and accessing link list structure. These functions are used by A* algorithm during its operation. The link lists are essentially circular link lists with a dummy node in order to ease the operation of insertion and deletion.
-operation 2
This section contains functions which implement the 8 puzzle moves e.g. finding the cost associated with the specified node, making legal moves, comparing two nodes for equality.
- operation 3
This section basically implements the A* algorithm.
- operation 4
This section involves interaction with the hardware.

The GUI in figures 5, 6 and 7 Shows the interface where the user can access the AS/RS to determine which pallet is occupied and which is free.

For the mechanism development, the components used are bevel gear, spur gear, rack and pinion, DC geared motor, solenoid and shaft. The A* algorithm for the sequence of moves is written in C language using custom computer service C compiler (ccsc) to store and retrieve specific pallet in and out of the storage retrieval system as requested.

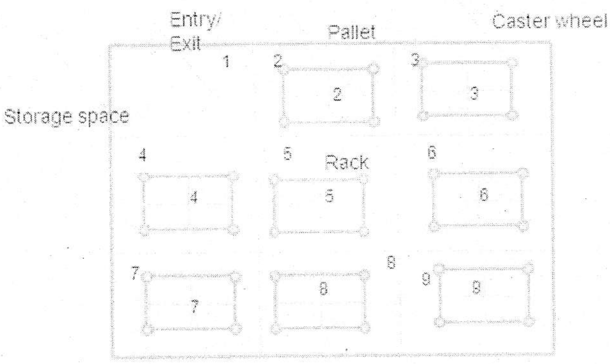


Figure 4. AS/RS Layout

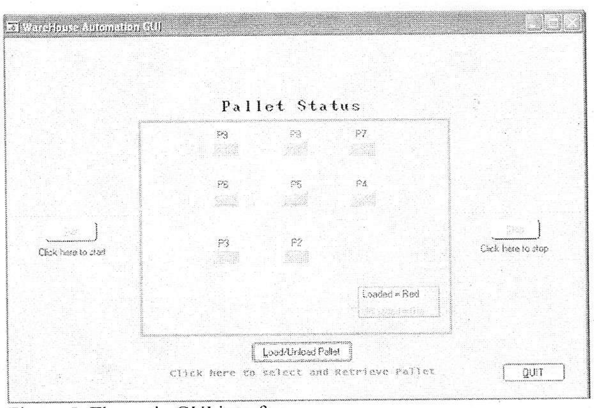


Figure 5, The main GUI interface

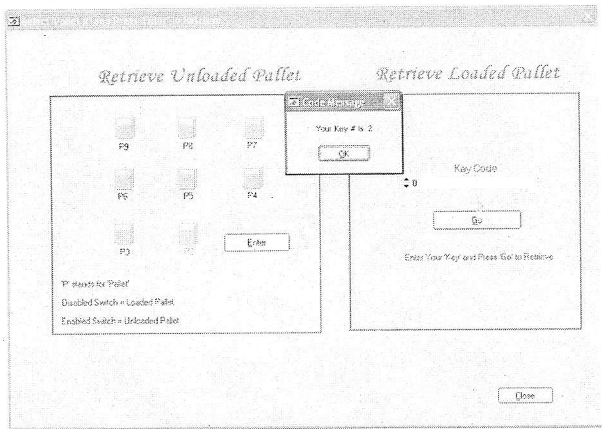


Figure 6, Interface for retrieving or storing a pallet

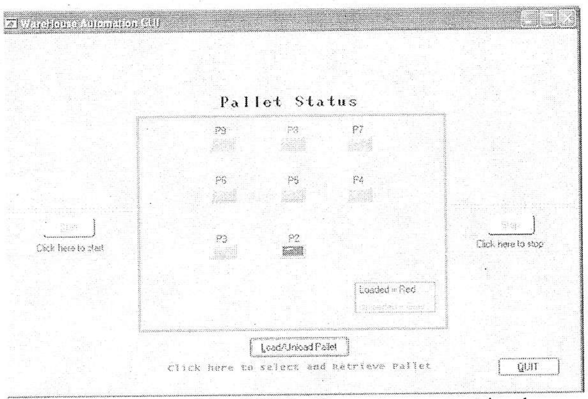


Figure 7, The status of the AS/RS after a storage or retrieval

## IV. RESULTS ANALYSIS

The default page of GUI in figure 5 shows the status of pallets either loaded or unloaded. If a user wants to access any pallet, Load/Unload pallet must be clicked and only those pallets that are available to users can be loaded and unloaded by them. Once the unloaded pallet is selected by the user, a 'Key' is provided which can be entered on the second box for loaded pallets to retrieve as shown in figure6.

As the user selects the pallet number to retrieve either loaded or unloaded, the pallet number is serially transmitted to SPI-Slave circuit. Slave places this code on data line and tells the master that code is available by setting the slave pin to high. Now as the master is notified to collect data, the SPI communication begins and master collects data. After collecting data master acknowledges the slave that data is collected by means of second handshaking signal 'ack'. As soon as master received pallet number, it causes the A* puzzle solving process to take place.

If a user wants to retrieve load on pallet 2, entering the desired pallet number, the pallet moves to the space 1 and space 2 becomes empty as shown in figure 8. It can be seen in microcontroller view in figure 10 that pallet 2 moves to the empty position and the empty space is now in pallet 2 space thus completing the cycle for retrieving pallet 2. To return pallet 2 back to its position another load has been placed on it, load switch changes status to ON to tell the microcontroller that another load has been placed. The sequence takes a reverse order back to their location on the storage space.

Another sequence this can take is not to sort the pallet retrieved back to its original position. In this case the new pallet to be retrieved by another user takes the present position of the AS/RS as the initial position and the storing and retrieving operation is done form this positions.

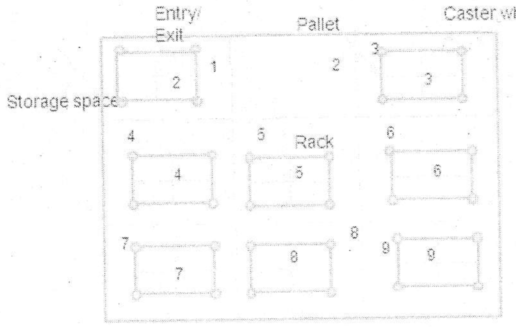


Figure 8, The AS/RS after retrieving a pallet

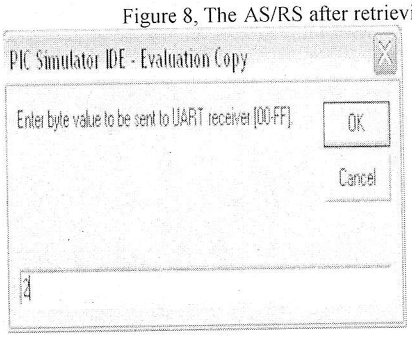


Figure 9, retrieving a pallet on simulator IDE

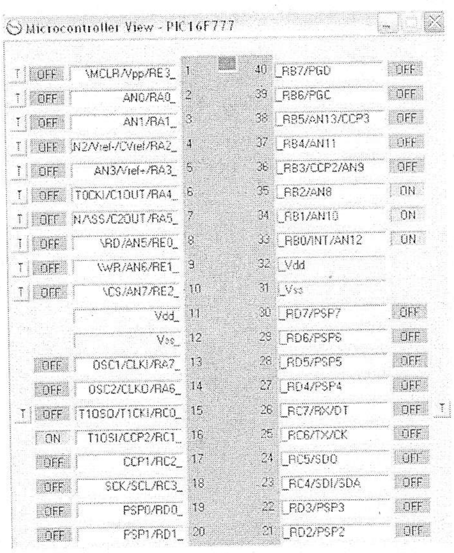


Figure 10, The movement of the Motor and Solenoid

## V. CONCLUSION

This research solves the problem of identifying and retrieving of pallets that changes address in course of time. This puzzle like automated storage and retrieval system is implemented to save space which does not require drive ways for storage and retrieval machine.

## ACKNOWLEDGMENT

The authors of this paper are grateful to the Research Management Centre, international Islamic university, Malaysia for funding this project under the research grant EDWB0701-08.

## REFERENCES

[1] Y. Chen and F. He, "Research on particle swarm optimization in location assignment optimization", in proc 7th world congress on int. contrl & automation, Chongqing, China, pp ,2008.

[2] S. S. Ho and S. Sarma, "The Fragmented Warehouse: Location Assignment for Unit-load Picking", Proceedings of the 2008 IEEE IEEM, 2008.

[3] http://www.mmh.com/article/CA6466042.html,Automated storage and retrieval system (AS/RS) basics, downloaded on 16th march 2009.

[4] M.H. Chih, YC Kai, Mu-CC, "Batching orders in warehouses by minimizing travel distance with genetic algorithms", Comp. in ind, Elsevier science, vol 56, pp. 169-178, 2005.

[5] M.P.Groover, Automation, production systems, and computer-integrated manufacturing, new York: Prentice Hall, 2001, Chapter 11,pp.

[6] X. Zhang, M.Tao, X.Han, "Optimizing Fixed Shelf Order-Picking for AS/RS Based on Immune Particle Swarm Optimization Algorithm", Proc. IEEE Int. Conf. on Auto & Logistics, China,2007.

[7] O.V.K. Chetty, M.S. Reddy, "Genetic algorithm for studies on AS/RS integrated with machines", Int J Adv Manuf Technol vol 22,pp. 932–940,2003.

[8] K. Yaghoub, W. Shouyang, "A Genetic Algorithm for Order Picking in Automated Storage and Retrieval Systems with Multiple Stock Locations", Vol. 4, No. 2, pp. 136-144, 2005.

[9] X. Zhang, X. Kong, X.Han X, "Modeling and Optimizing Fixed Shelf Order-Picking for AS/RS Based on Least Time", Proc. IEEE Int. Conf on Auto & Logistics, China, 2008.

[10] P.V.D.B Jeroen, A.J.R.M Gademann, "Simulation study of an automated storage/retrieval system", int. j. prod. Res., vol.38, no 6, pp.1339-1356, 2000.

[11] ] Y. Jih, Y. Yuehwern, "Using neural networks to select a control strategy for automated storage and retrieval systems (AS/RS)", int. j. comp. integ. Manuf. vol. 10, no. 6, pp. 487- 495, 1997.

[12] Y. Hu, X. Xu, W. Hsu, A. Toh, L. Cheong, K. Chee, T.Song, "Efficient Algorithms for Load Shuffling in Automated Storage and Retrieval Systems",Proc IEEE Int. Symp on Comput. Intelli. in Robotics and Auto. , July 2003.

[13] Y. Bo, D. Lee, "AS/RS simulation and optimization based on flexsim", 978-1-4244-3894-5/09, IEEE,2009.

[14] M. Fogel, B. Nathan, H. Ren, J. Schiff, M. Mac, G. Ken, "Automated tracking of pallets in warehouses", proc 3rd IEEE conf. on Auto. Sci & engrg., USA, 2007.

[15] A. Kenneth, Berman and L.P. Jerome, Algorithms: Sequential, Parallel, and Distributed, Boston: Thomson, 2005, chapter23, pp 715-743

[16] S. Onut, U.R. Tuzkaya, B. Dogac, " A particle swarm optimization algorithm for the multiple-level warehouse layout design problem", Comp. in ind, Elsevier science, vol. 54, pp. 783-799, 2008.