

A Cognitive Robotic System for a Human-Following Robot

Chien Van DANG^a, Heungju AHN^b, Hyeon C. SEO^a, and Sang C. LEE^{a,1}

^a*IO Lab, Division of Intelligent Robot, Convergence Research Institute, DGIST, Daegu, 42988, Rep. of Korea*

^b*College of Transdisciplinary, DGIST, Daegu, 42988, Rep. of Korea*

Abstract. In this paper we propose a cognitive robotic system that utilizes computational psychology (the Soar cognitive architecture) and an obstacle avoidance method (modified dynamic window approach) in ROS (Robot Operating System) platform for controlling a mobile robot. This system is applied to perform a task of human-following, aiming to help the robot navigate itself to the target person avoiding collision. A cognitive agent based on Soar cognitive architecture is created to reason its current situation and make decisions on movement direction such as go-straight, turn-left or turn-right, whereas the dynamic window approach is modified to avoid collision by computing appropriate velocities for driving the robot motors. To the end, a part of implementation is presented to describes how the system works.

Keywords. Cognitive robotic system, obstacle avoidance, following mobile robot

1. Introduction

The problem of human-following for mobile robots has been attracted by world-wide researchers over twenty years. There are two main major issues: firstly, detecting and determining distance from the robot to the target person; secondly, navigating the robot to the target user. In order to achieve such problem, there are a number of different approaches. In particular, we can mention approaches that are using vision based techniques [1,2], non-vision based ones [3,4], transmitter-receiver based ones [5,6], intelligent space techniques [7], and multi-modal approaches [8,9].

This research belongs to multi-modal approach. In order to determine distance from the robot to the target, we use a local positioning system with ultra wideband sensors: a transmitter carried by the target person transmits ultra wave signals, two receivers mounted on the robot receive the signals [10]. However, as the second issue, on which this paper focuses, the robot navigating to the target person needs more attention in unstructured environments. A cognitive robotic system, which associates computational

¹Corresponding Author (E-mail:sclee@dgist.ac.kr)

This work was partly supported by DGIST R&D program of the Ministry of Science, ICT, and Future Planning of KOREA (20-IT-03) and Korea Institute of Energy Technology Evaluation and Planning (KETEP) grant funded by the Korea government (MOTIE) (2019-3010032490, Development of 25kW modular fuel cell power conversion system)

psychology based on Soar cognitive architecture with ROS platform, is utilized. Possessing effective knowledge representations with both short-term and long-term knowledge for accessing memory, learning and decision-making, the architecture aims to human-like intelligence [11,12]. On the other hand, ROS platform supports mechanisms for the system to be able to communicate easily with devices. A Soar based agent illustrates how the robot should behave heading to the human target. Besides, a modified dynamic window approach is also proposed as a local controller for the robot.

The rest of this paper is organized as follows. Section 2 describes hardware architecture of a testbed robot and its motion equations with differential drive. Section 3 presents our cognitive robotic system which contains a Soar cognitive agent and a local controller using modified dynamic window approach. A data graph of the system is presented as well. And we wrap-up with discussion and further work in section 4.

2. Robot hardware architecture and motion equations

2.1. Robot hardware architecture

A testbed robot used in this research includes an embedded motor driver board, an embedded control board, three ultra wideband sensors (two on robot, one is carried by a human user), and an RGB-D camera as shown in **Figure 1**. The motor driver board has multiple input and output capabilities, including direct current motors, encoders, and a serial port. The control board, UP squared, with an Intel Atom x7-E3950 2.0GHz processor and 4GB of RAM using the Linux Ubuntu operating system is used to communicate with the motor driver board and several sensors. Working on ROS platform, driver nodes of these devices can be able to easily exchange information. A Soar package is also built on this control board for creating an agent to control the system. Ultra wideband (UWB) sensors are Decawave DWM1001 modules which are claimed to measure distance with a 10cm degree of accuracy [13]. For measuring distance from the robot to the target user, two UWB modules are mounted on the top front left and right of the robot, and one UWB module is carried by the user. An RGB-D camera is an Intel Realsense D435 depth camera which is used to sense obstacles in surrounding environment.

In addition, other hardware components include a DC-DC regulator which converts the battery power 24 voltages to 5 voltages to power the UP board. A monitor screen is also included for monitoring during the debugging process.

2.2. Motion equations for a differential robot

The mobile robot used in this research is equipped with differential drive. That means that the direction of motion is controlled by separately controlling speeds v_l and v_r of the left and right wheels respectively. Starting from the pose $[x, y, \theta]$ at the time t , this motion modeling determines the new pose $[x', y', \theta']$ at the time $(t + \delta t)$ given the control v_l and v_r .

First of all, when the robot is moving in a curve, there exists a center of that curve at that moment, known as the Instantaneous Center of Curvature (ICC). Let R be the radius of the curve, and L be the distance between the two wheels. Then the rate of rotation ω around the ICC is related to the velocity of the robot by equations:

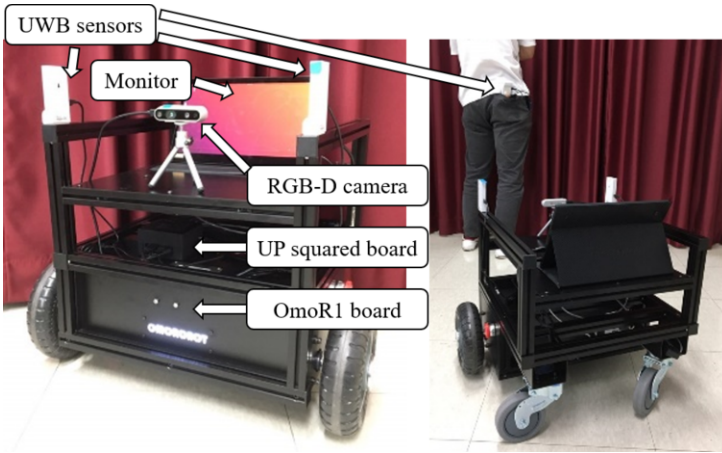


Figure 1. A testbed robot and its components

$$\omega \left(R + \frac{L}{2} \right) = v_r, \quad \text{and} \quad \omega \left(R - \frac{L}{2} \right) = v_l \tag{1}$$

These equations give values of R and ω :

$$\omega = \frac{v_r - v_l}{L}, \quad \text{and} \quad R = \frac{L \cdot (v_r + v_l)}{2(v_r - v_l)} \tag{2}$$

The angle that the robot rotates with angular velocity ω in period of time δt is expressed as follows:

$$\omega \delta t = \frac{(v_r - v_l) \delta t}{L} \tag{3}$$

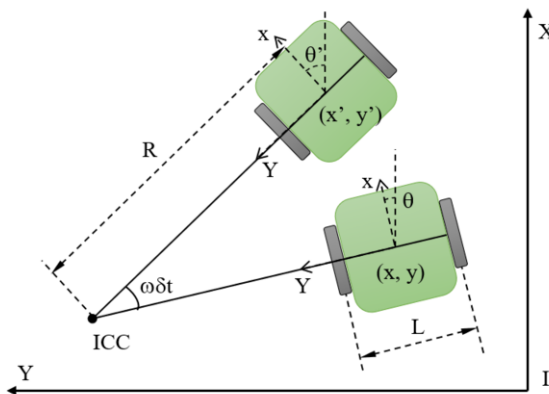


Figure 2. The robot rotating around ICC

We will present the pose of the robot a vector $[x, y, \theta]^T$, where (x, y) is a coordinates of the position of the robot in the xy -plane, θ is the heading which points in the forward

direction of the robot. **Figure 2** illustrates relative position between the current pose and the next pose of the robot. To retrieve the next pose, the robot performs a consequent steps: start at the initial frame, translate out to the current position (${}^I T_c$); rotate to the current orientation (${}^c R_c$); translate to the ICC position (${}^c T_{icc}$); rotate around the ICC to the new angle $\omega\delta t$; finally, translate out to the new position. These steps are described by the equation:

$${}^I T_N = {}^I T_c \cdot {}^c T_{c\theta} \cdot {}^c T_{icc} \cdot {}^{icc} T_{\omega\delta t} \cdot \omega\delta t T_N \quad (4)$$

Equivalently

$${}^I T_N = \begin{bmatrix} 1 & 0 & 0 & x \\ 0 & 1 & 0 & y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \cos \theta & -\sin \theta & 0 & 0 \\ \sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & \theta \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & R \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \cos \omega\delta t & -\sin \omega\delta t & 0 & 0 \\ \sin \omega\delta t & \cos \omega\delta t & 0 & 0 \\ 0 & 0 & 1 & \omega\delta t \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -R \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Equation (4) results a set of new pose of the robot as follows:

$$\begin{bmatrix} x' \\ y' \\ \theta' \end{bmatrix} = \begin{bmatrix} R \cos \theta \sin \omega\delta t + R \sin \theta \cos \omega\delta t + x - R \sin \theta \\ R \sin \theta \sin \omega\delta t - R \cos \theta \cos \omega\delta t + y + R \cos \theta \\ \theta + \omega\delta t \end{bmatrix} \quad (5)$$

After getting motion mode of the robot, we start to implement a controller on the robot as presented in the next section.

3. Cognitive robotic system

3.1. Cognitive agent

In our cognitive robotic system, a cognitive agent based on Soar cognitive architecture is implemented. A Soar agent attempts to use its knowledge in form of symbol representations about its current state, after applying operation, which will be changed into next state in an effort to reach its goal. The current state which includes the state of the agent and its environment is stored in working memory, that can be regarded as a short-term memory. In the agent, long-term knowledge is encoded as production rules. These rules consist of elaborations and operators to add information and make changes to working memory. Detail information about Soar can be found in [11].

As production rules, operators can be organized into problem spaces that are relevant to specific problem. In this research, the agent has a problem space called avoid-obstacle, in which operators describe how it should move if an obstacle is sensed.

Figure 3 shows a symbol graph of working memory in our following agent. The graph represents relation between the mobile robot, the user and obstacle, as well as their property characteristics. Labels in circles (S1, O1, R1, U1, P1, P2) are called identifiers. Words preceded by a character ^ (robot, user, obstacle, so on) are called attributes. "Combo", "David", "Obs1", and numbers in float type are values that are constant, whereas U1 is the value associated with the character ^ preceding attribute, but is an identifier instead of a constant.

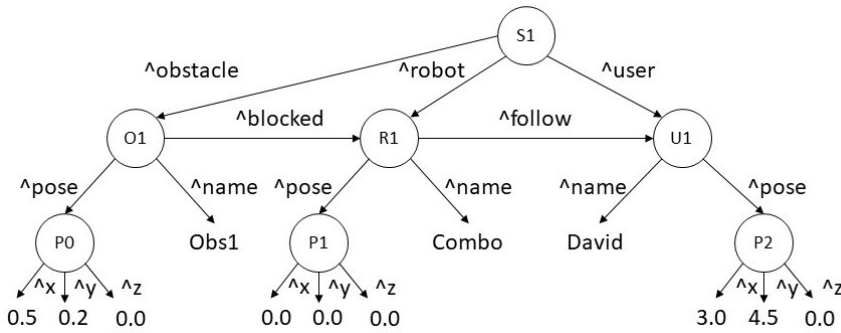


Figure 3. A part of working memory in our cognitive agent

A production rule for creating an operator can be briefly written as follows: If the agent is blocked by an obstacle, the obstacle is on the right direction, then an operator of action turn-left is proposed. After that, another production rule will apply the operator of turn-left and make changes to working memory. A number of actions are used for the agent to make change to the environment such as go-straight, turn-left, turn-right. These actions are then executed by a local controller using a modified version of dynamic window approach which is presented in the next section.

3.2. Modified dynamic window approach

Algorithm 1 MDWA(robot, configure, goal, obstacle, direction)

```

1:  $allowable\_V \leftarrow generateWindow(robot, configure, direction)$ 
2:  $allowable\_W \leftarrow generateWindow(robot, configure, direction)$ 
3: for each  $v$  in  $allowable\_V$  do
4:   for each  $w$  in  $allowable\_W$  do
5:      $trajectory \leftarrow predictTrajectory(v, w)$ 
6:      $heading \leftarrow headingCost(trajectory, goal)$ 
7:      $clearance \leftarrow clearanceCost(trajectory, obstacle)$ 
8:      $speed \leftarrow speedCost(v)$ 
9:      $cost \leftarrow costFunction(heading, clearance, speed)$ 
10:    if  $cost > optimal$  then
11:       $best\_v \leftarrow v$ 
12:       $best\_w \leftarrow w$ 
13:       $optimal \leftarrow cost$ 
14:    end if
15:  end for
16: end for
17:  $velocity\_command \leftarrow (best\_v, best\_w)$ 

```

Obstacle avoidance controller plays an important role to help the robot tackle current situation while operating in real environment. In this research, a modified dynamic window approach (MDWA) algorithm is utilized for the robot to avoid both static and dynamic obstacles. Pseudo code of the MDWA is described above.

This modified version of dynamic window approach differs from the original version at a point that the movement direction to goal is decided by a cognitive agent. Directions commanded from the agent is converted into a window of velocities in form of $[V_{\min}, V_{\max}, W_{\min}, W_{\max}]$, where V_{\min}, V_{\max} are minimum and maximum translational velocities, W_{\min}, W_{\max} are minimum and maximum angular velocities respectively. For example, turn-left direction refers to a velocity window of $[0, V_{\max}, 0, W_{\max}]$. This window is added to generate velocity window to specify a possible set of velocities for the robot.

3.3. Data flow of the system

During execution of the robot, the following five ROS nodes are concurrently running:

- (1) Detect and determine distance from the robot to the target,
- (2) Measure distance from the robot to its surroundings,
- (3) Acquire motor states,
- (4) Interface with Soar and send output commands for control robot behaviors,
- (5) Compute translational and angular velocities to drive motors.

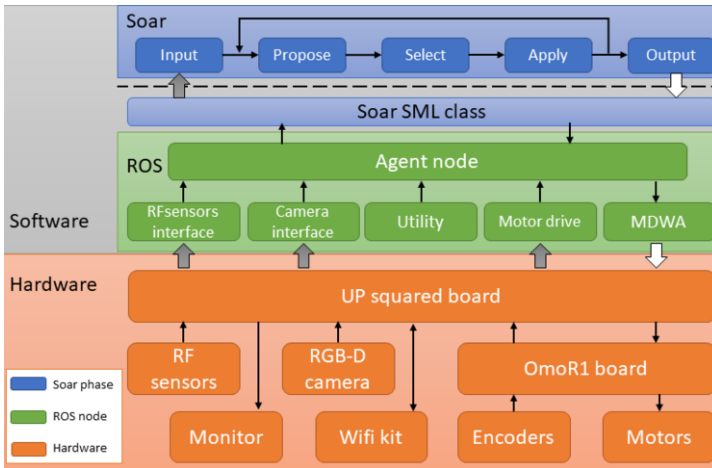


Figure 4. Data flow graph between software and hardware

Figure 4 shows the data flow between hardware devices and software components in the system. The hardware is shown on the bottom layer including an UP squared board, an Omor1 board, and several devices: ultra wideband sensor, RGB-D camera, motors with encoders. The software in the CRS has primarily two parts: Soar, ROS. The main node, agent node, coordinates communication between five other nodes, all in ROS environment: RFsensor interface node, camera interface node, utility node, motor drive node, and MDWA node. The RFsensor node is a program that activates two UWB sensor modules at the same time, receives measured distance values, then computes relative position and angular direction of the target, and finally sends them to the main node for further processing. The camera node subscribes images from the depth camera, and publishes 3-dimensional coordinates of obstacle.

Besides, the motor drive node is activated to bring-up the robot. Last but not least, the MDWA node is the modified dynamic window approach in order for the robot to be able to avoid static and dynamic obstacles. The agent node is a node that subscribes sensor data, sends them to Soar agent via input links, receives behavior commands from the agent via output links, then publishes commands to the MDWA node to activate the robot.

4. Discussion and future work

This paper introduces a cognitive robotic system which associates Soar cognitive architecture with an obstacle avoidance method in ROS platform in order to operate a mobile robot following the human user. Two main parts including a cognitive agent and a obstacle avoidance controller are described. The agent attempts to reason its current situation, and make decisions on direction of movement, whereas the controller using a modified version of dynamic window approach computes a set of translational and angular velocities to drive the robot motors. Then data flow of the system is presented to show how the system works. Among general approaches to accomplish a human-following robot, this approach of using cognitive robotic system is newly proposed offering the robot abilities of reasoning and decision making like humans to avoid obstacle while following the human user. In additions to local positioning problem, we also have to deal with knowledge representations which should be further studied in the future.

References

- [1] Dixon W.E., Dawson D.M., Zergeroglu E., Behal A. Adaptive tracking control of a wheeled mobile robot via an uncalibrated camera system. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*. 2001 Jun; vol. 31(3): p. 341-352.
- [2] Emina P., Adrian L., Danijela R.D., Vlastimir N. Stereo vision-based human tracking for robotic follower. *International Journal of Advanced Robotic Systems*; 2013 Feb; vol 10(5).
- [3] Kawarazaki N., Kuwae L.T., Yoshidome T. Development of human following mobile robot system using laser range scanner. *Procedia Computer Science*. 2015; vol 76: p. 455-460.
- [4] Peng W., Wang J., Chen W. Tracking control of human-following robot with sonar sensors. *Intelligent autonomous systems*. 2016 Feb. vol 531: p. 301-313.
- [5] Akihisa O., Yousuke N., Youhei G. Intelligent escort robot moving together with human – Methods for human position recognition. 2002 Jan.
- [6] Tao F., Yao Y., Lin W., Yanru B., Ziang X., Zhen L. A human-tracking robot using ultra wideband technology. *IEEE Access*. 2018 July; vol. 6: p. 42541-42550.
- [7] Morioka K., Lee J-H., Hashimoto H. Human-following mobile robot in a distributed intelligent sensor network. *IEEE Transactions on Industrial Electronics*. 2004 Feb. vol 51(1): p. 229-237.
- [8] Algabri R., Choi M-T. Deep-learning-based indoor human following of mobile robot using color feature. *Sensors*. 2020 May; vol 20: p. 2699-2718.
- [9] Sebastian L., Marcus K., Sascha H., Jannik F., Gernot A.F., Gerhard S. Providing the basis for human-robot-interaction: a multi-modal attention system for a mobile robot. In *Proceedings of the 5th International conference on Multimodal interfaces (ICMI '03)*. 2003 Nov. p. 28-35.
- [10] Sang C.L., Syamsul R., Heungju A. Anchor placement design for a decoupled simple trilateration algorithm. *Journal of Nonlinear and Convex analysis*. 2019 July; vol 20(7): p. 1327-1339.
- [11] Laird J. *The Soar cognitive architecture*. The MIT Press. 2012.
- [12] Laird J. et al. "Cognitive robotics using the Soar cognitive architecture. *CogRob@AAAI*. 2012.
- [13] Decawave DWM1001 development module website: <https://www.decawave.com/product/dwm1001-development-board/> (available on 30th Aug. 2020).