# Context-aware Decision Making for Maze Solving

Sheir Afgen Zaheer and Jong-Hwan Kim

Department of Electrical Engineering, KAIST, 291 Daehak-ro,
Yuseong-gu, Daejeon, Republic of Korea
{sheir, johkim}@rit.kaist.ac.kr

**Abstract.** This paper proposes a context-aware decision making framework for a maze solving robot. The proposed architecture utilizes a fuzzy integral based decision making scheme to select the best behavior according to the current environmental context of the robot. The simulation results show that despite having no prior information about the arrangement of the maze, the proposed cognitive architecture for context-aware decision making successfully enabled the robot to find its way through the maze.

**Keywords:** Fuzzy integral, Multi-criteria decision making, Maze Solving Robot

## 1 Introduction

Over the past few decades, cognitive decision making in intelligent robots and artificial agents have been a popular research area. The human cognitive intelligence seems to be the main inspiration behind all this interest. However, the functional architecture of human cognitive process is still not an established science; this leads to a variety of interpretations and understandings about its functionality. Some researchers are inclined towards symbolic-based cognitive architectures [1]-[2], while others pursue the development of emergent architectures [3]. There is some research on hybrid architectures as-well [4], these architectures result from the combination of both the symbolic and the emergent schemes. Our work is mainly inspired from some of the more recent researches that use personal and environmental aspects (or symbols) as criteria for cognitive decision making [5]-[6].

This project proposes a framework based on context-aware decision making for a maze solving robot. The problem setup consists of a simulated robot, spawned at the starting point of a maze. The maze is a square-shaped arena with randomly placed walls (Fig. 1). The robot is spawned at the starting position of the maze without any information about the map of the maze. Therefore, the robot needs to solve the maze based on its real time interaction with the environment inside the maze. The location of the destination point is the only prior information that the robot has. Though there have been some previous researches on maze solving by robots, most of them use a run or two to map the environment inside the maze before coming up with the solution path [7]-[8]. Another research was aimed at similar kind of cognitive architecture for maze solving [9]. The robot in this research, however, had an additional benefit of being able to look at maze from the top of the maze's walls. The core of the proposed context-aware decision making is a two criteria decision process. Choquet fuzzy integral has been

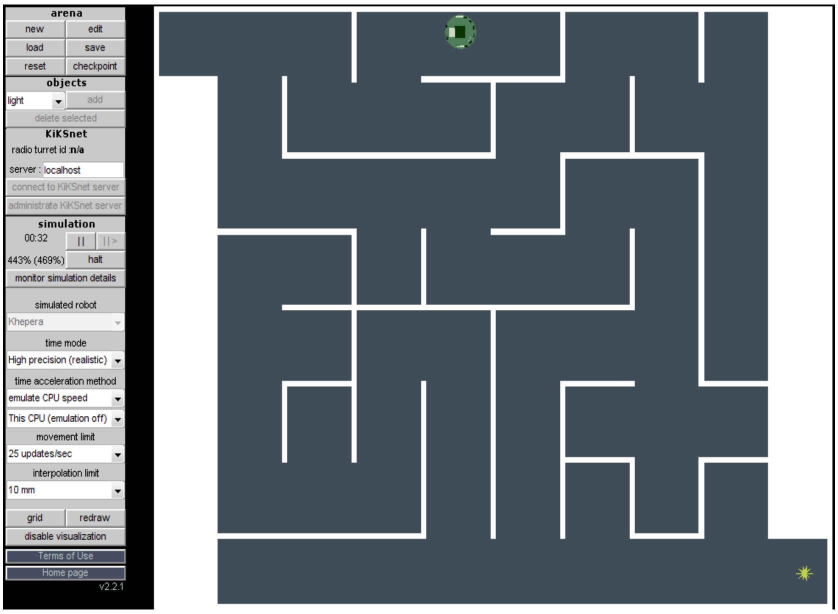



**Fig. 1.** A Snapshot of simulated maze.

found to be an effective aggregator for such decision making problems [6]. In our work, we have used KiKS (a Khepera robot simulator) [10] for simulating Khepera robot and mazes. Khepera is a small differential wheeled robot with an array of ultrasonic sensors attached to its circular periphery.

This paper is organised as follows: Section 2 explains the proposed framework for context-aware decision making to solve a maze. Section 3 describes the simulations results. Finally, discussions and conclusions follow in section 4.

## 2 Decision Making Framework for The Maze Solving Robot

This paper proposes a decision making framework for a mobile robot that enables it to find its way through an unknown maze. The overall architecture of the proposed framework is summarized in Fig. 2. This framework consists of four layers: the perception layer, the reasoning layer, the memory layer and the execution layer. Each layer consists of some modules through which it performs its share of tasks. Following subsections provide the functional description of all these layers and modules.

### 2.1 The Perception Layer

The perception layer is mainly responsible for interpreting the sensory readings and transforming them into meaningful information. The components of this layer are the array of sensors on the robot and the perception module.

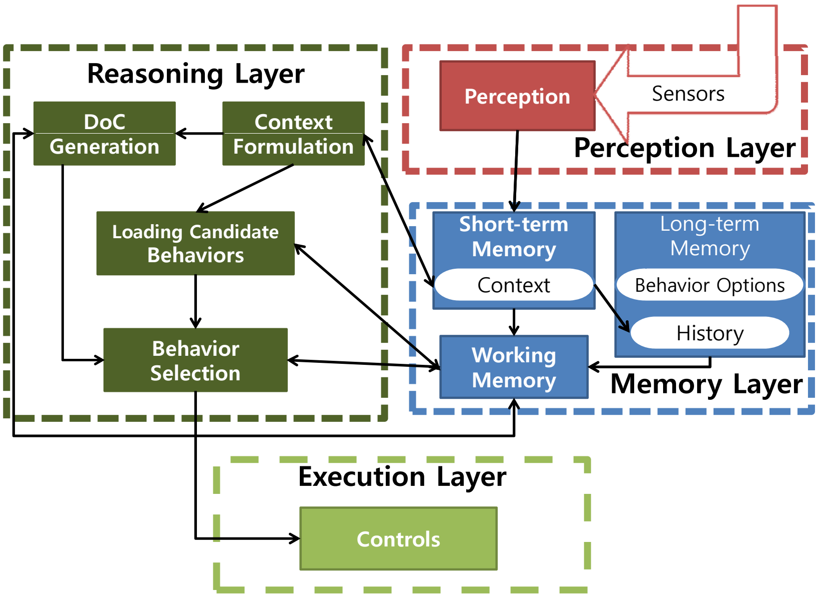



**Fig. 2.** The architecture of the decision making framework for the maze solving robot.

**Sensors:** As mentioned before, the robot considered in this setup has an array of eight sensors along its periphery. All these sensors are identical ultrasonic distance sensors. These sensors provide a 10-bit proximity reading, i.e. from 0-1024; where 1024 implies a physical contact. In addition to that there are two wheel encoders as-well. These encoders can be used to obtain the information about the robot's heading and the distance travelled.

**The perception module:** Based on readings provided by the sensors, the perception module receives an array of ultrasonic sensor readings and the odometric readings from the encoders. The perception module is responsible for meaningful interpretation of this information. First, it translates the proximity readings into the free space available to the left, right and in front of the robot. Secondly, it uses the odometric information to determine the heading of the robot and its position relative to the starting point of the maze.

### 2.2 The Memory Layer

As the name suggests the memory layer is responsible for information storage. Based on the type of information stored, the memory layer can be divided into three components: short term memory, long term memory and working memory.

**The short-term memory:** The short term memory, in this case, is responsible for the storage of current context only. The short term memory interacts with the perception layer and the context formulation module in the reasoning layer to perform its functionality.

**The long-term memory:** The long term memory stores two types of information: behavior options and history. The behavior options module stores the list of possible behaviors associated with each context situation. (For details on this, refer to section 2.3) The history modules stores the past context information, in order to keep track of the previously visited sites inside the maze. This particular information is used by behavior selection module.

**The working memory:** This particular part of the memory is responsible to hold information that is being used in the on-going decision making, including the history and the behavior options, whenever required by the reasoning layer.

### 2.3 The Reasoning Layer

The reason layer is the core of the proposed framework for context-aware decision making. This is where the intelligent decision making stems from. As shown in Fig. 2, the reasoning layer consists of four phases: Context formulation, Degree of consideration (DoC) generation, loading candidate behaviors and Behavior Selection. All these phases are explained in the following subsections.

**Context formulation:** As the name suggests, this phase is responsible for building the context. The built context in this application setup consisted of three information variables: position of the robot, heading of the robot and the current local situation. The list of local situations and their description is provided in Table 1 (first two columns).

**Degree of consideration generation:** As mentioned in the previous section, there are two criteria for determining the output behavior for the robot. These two criteria are:

i) The change in the Euclidean distance to destination ($\varDelta D$)
ii) Distance to the next obstacle in that direction ($d_o$)

This phase of the decision making process is responsible for the generating the DoCs for both these criteria. The DoCs, in this case, are represented by $\lambda$-fuzzy measures. Since there are only two criteria, $\varDelta D$ and $d_o$, their union can be written as [11]:

$$g(\varDelta D \cup d_o) = g(\varDelta D) + g(d_o) + \lambda g(\varDelta D) g(d_o) = 1, \quad (1)$$

where $g(.)$ represents the $\lambda$-fuzzy measure for a criterion or a set of criteria, and $\lambda$ is the interaction degree index. However, in this case, the criteria are considered to be uncorrelated. The justification for this consideration is based on the fact that an increase

**Table 1.** List of local situations and corresponding Candidate behaviors

| Local situation | Description | Candidate behaviors |
|---|---|---|
| Backwards only | The robot can only go back, it's a dead end | 180 degree turn |
| One way Forward | Straight path with walls on left and right | Go straight |
| One way Left | Only open area is towards left | Turn left |
| One way Right | Only open area is towards right | Turn right |
| Two way Forw/Left | A junction with open areas in forward and left directions | Go straight; Turn left |
| Two way Forw/Right | A junction with open areas in forward and right directions | Go straight; Turn right |
| Two way Left/Right | A junction with open areas in right and left directions | Turn left; Turn right |
| Three way | A '+' like junction where the robot can go in an any direction | Turn left; Turn right; Go straight |

or decrease in $\varDelta D$ has no effect on $d_o$ and vice versa. Hence, $\lambda$ is zero in this case, and (1) simplifies into:

$$g(\varDelta D \cup d_o) = g(\varDelta D) + g(d_o) = 1. \tag{2}$$

The next step is to define the DoC values, i.e. the fuzzy measures, for both criteria. However, the DoC values in this case are not constant. They vary as the robot moves around. The DoC value for $\varDelta D$ is defined as a function of the absolute distance from the destination position before the current behavior selection.

$$g(\varDelta D) = g_{max} - (\frac{D_c}{D_T} \times (g_{max} - g_{min})), \tag{3}$$

where $g_{max}$ and $g_{min}$ are the maximum and minimum permissible values for the DoC of $\varDelta D$. $D_c$ is the current euclidean distance of the robot from the destination point, and $D_T$ is the diagonal length of the maze. From (2), the DoC for $d_o$ can be defined as:

$$g(d_o) = 1 - g(\varDelta D). \tag{4}$$

**Loading Candidate Behaviors:** In this problem setup, all robot behaviors are not applicable for each context. Therefore, depending on the current context, a list of possible behaviors needs to be provided. Possible contexts and the associated behaviors are summarized in Table 1 . The information in this table is available in behavior options module of the long term memory. This phase of decision making receives the context from the context formulation and then, loads the candidate behaviors from the long term memory through working memory.

**Behavior Selection:** The final step in the reasoning layer was to select the best behavior from the loaded candidate behaviors. Now that all the DoC values and the candidate

**Table 2.** Lookup table for partial evaluation values over $\Delta D$, i.e. $h(\Delta D)$

| **Change in distance along axis** | Horizontal axis | | |
|---|---|---|---|
| Vertical axis | Positive | Zero | Negative |
| Positive | 0.05 | 0.25 | 0.5 |
| Zero | 0.25 | Not applicable | 0.75 |
| Negative | 0.5 | 0.75 | 0.95 |

behaviors are available, an aggregation operation for global evaluation is required. The aggregator used here is the discrete Choquet fuzzy integral [12].

$$\int_X h \circ g = \sum_{i=1}^{n} (h(x_i) - h(x_{i-1}))g(E_i), \tag{5}$$

where $n$ is the number of criteria, $h(.)$ is the partial evaluation value, and $E_i$ is the subset of the criteria set $X$ consisting $x_i$ and all others that have a higher partial evaluation value than $x_i$. In this application, there are two criteria, $x_1 = \Delta D$ and $x_2 = d_o$. Therefore, before evaluating each candidate behavior, partial evaluation values over each criteria, $h(.)$, are required. For the first criterion, i.e. $\Delta D$, the partial evaluation for a particular candidate behavior over $\Delta D$ is obtained from a lookup table (Table 2 ). As shown in the table, the partial evaluation value depends on whether the selected behavior results in an increase or decrease in Eucledian distance to the destination point. The partial evaluation over the second criteria, i.e. $d_o$, is defined as a function of proximity of the next obstacle in the direction that results from the current candidate behavior. Numerically, it is given as:

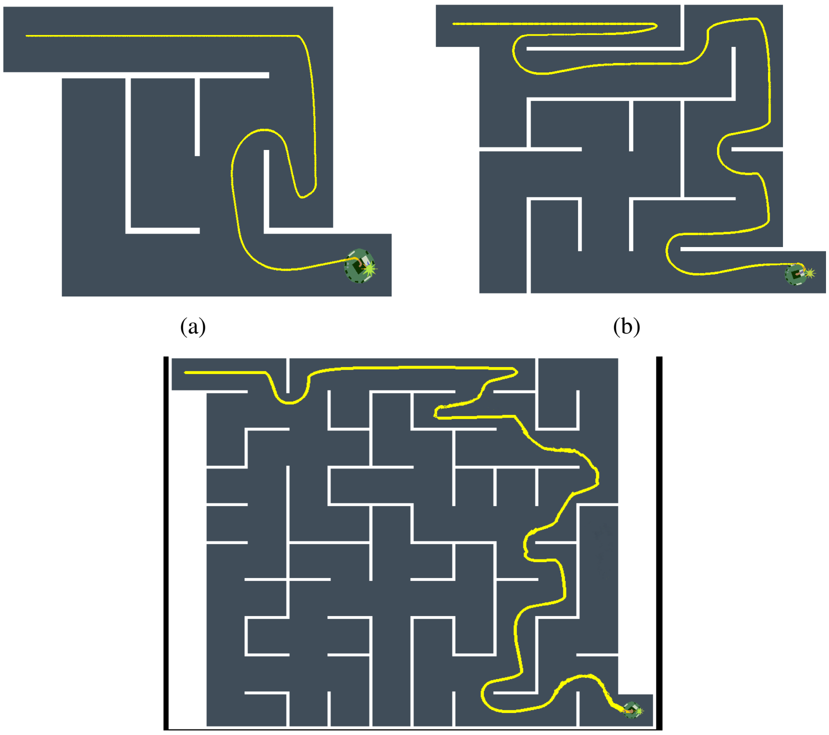
$$h(d_o) = 1 - \frac{PV_c}{PV_{max}}, \tag{6}$$

where $PV_c$ is the current proximity value measured by the ultrasonic sensor, and $PV_{max}$ is the maximum proximity value when the sensor is touching the obstacle.

Finally, the candidate behavior with the highest global evaluation value from (5) is selected as the winning behavior. However, this behavior is not sent directly to the execution layer. Instead, the history in the long term memory is first checked. If the history tells that the current context, including the robot position, has already been visited, it means that the previously selected behavior at this particular location of the maze was not correct. If such a case arises, the behavior with the next highest evaluation value is selected. This helps the robot to avoid making the same mistake.

### 2.4 The Execution Layer

The execution layer is responsible to ensure proper implementation of the behavior selected through the reasoning process. The control module in this layer translates the behavior command into the respective motor velocities.



(a) (b) (c)

**Fig. 3.** Robot's trajectory (a) 40 x 40 cm maze (b) 60 x 60 cm (c) 100 x 100 cm.

## 3 Results

Once, the cognitive architecture for maze solving was designed, the next step was to test it. Different simulated mazes were constructed for this purpose. Three mazes that vary in size and complexity are discussed here.

i) 40 x 40 cm maze
ii) 60 x 60 cm maze
iii) 100 x 100 cm maze

The resulting trajectories of the robot, while attempting to solve these mazes with the proposed architecture, are shown in Fig. 3. It is evident from the figure that the robot was able to successfully reach the destination point. However, we can easily see that the proposed decision making framework does not guarantee an optimal path. This limitation can be attributed to the fact that the robot's instantaneous knowledge about the structure of the maze is limited to a small region of maze that lies within the range of its sensors. Therefore, the decision made by the proposed decision making scheme, though the most viable decision within the visible context, may lead the robot to a dead end later on.

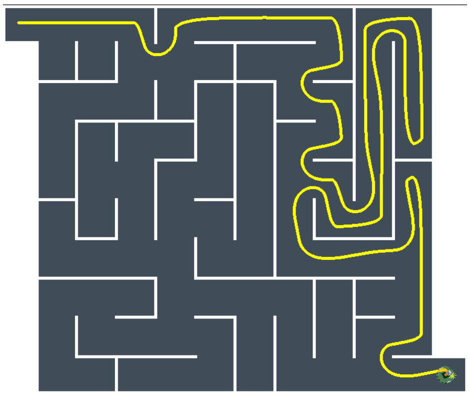


**Fig. 4.** Robot's trajectory for wall following algorithm

**Table 3.** Comparative results for the context-aware decision making algorithm versus the wall following algorithm

| Maze Size | CDM (seconds) | WF (seconds) |
|---|---|---|
| 40 x 40 | 97 | 123 |
| 60 x 60 | 152 | 187 |
| 100 x 100 | 203 | 314 |

Moreover, we also compared our algorithm with wall follower algorithm for maze following robot. We selected this algorithm for comparison because unlike other algorithms for maze solving robots, the wall following algorithm does not require a pre-run to map the maze. Trajectory result for one run of the wall following algorithm is shown in Fig. 4. It can be seen that the wall following algorithm just tracks the wall until it eventually leads to the end of the maze. The comparative results for the context-aware decision making (CDM) algorithm versus the wall following (WF) algorithm are shown in Table 3. The table shows the average maze solving time over 10 randomly generated mazes. The results show that the proposed algorithm performed better then the wall following algorithm.

## 4 Discussion and Conclusion

This paper proposed a cognitive architecture for a mobile robot that enabled it to find its way through the maze. The results show that the robot with the proposed cognitive architecture for behavior selection was able to navigate its way through the maze. Simulations with different maze sizes were conducted. Though the robot was successful in finding the destination point through the maze, it did show some tendency to make suboptimal decisions. In summary, the proposed method for maze solving leads to the successful solution but, in its current form, it cannot be claimed as the best or the optimal method. This work is in its preliminary stages yet. In future, we aim at improving the decision making process by including more context information and criteria. Moreover, we also aim to incorporate a learning module to enable the robot to learn and re-adjust the parameters of DoC generation for different criteria.

## Acknowledgment

This research was supported by the MKE (The Ministry of Knowledge Economy), Korea, under the National Robotics Research Center for Robot Intelligence Technology support program supervised by the NIPA (National IT Industry Promotion Agency) (NIPA-2010-N02100128).

## References

1. Laird JE, Newell A, Rosenbloom PS (1987) SOAR: An architecture for general intelligence. Artif Intell. 33(1):1-64
2. Kieras D E, Meyer DE (1997) An overview of the EPIC architecture for cognition and performance with application to human-computer interaction. Hum-Comput Interact. 12(4):391-438
3. Brooks R (1986) A robust layered control system for a mobile robot. IEEE J Robot Autom. 2(1):14-23
4. Bughart C et al (2005) A cognitive architecture for a humanoid robot: a first approach. Paper presented at IEEE-RAS international conference on humanoid robots, 5 Dec 2005
5. Kim J-H, Cho S-H, Kim Y-H, Park I-W (2008) Two-layered confabulation architecture for an artificial creature's behavior selection. IEEE Trans Syst Man Cybern C, Appl Rev. 38(6):834-840
6. Kim J-H, Ko W-R, Han J-H, Zaheer SA (2012) The degree of consideration-based mechanism of thought and its application to artificial creatures for behavior selection. IEEE Comput Intell Mag. 7(1):49-63
7. Jiang H, Wang H, Tian Y (2011) Design and realization of a maze robot. Presented at international conference on Consumer Electronics, Communications and Networks, 16-18 April 2011
8. Dang H, Song J, Guo Q (2010) An Efficient Algorithm for Robot Maze-Solving. Presented at 2nd international conference on Intelligent Human-Machine Systems and Cybernetics, 26-28 Aug 2010
9. Wyeth G, Browning B (1998) Cognitive Models of Spatial Navigation from a Robot Builders Perspective. Adaptive behavior. 6.3-4
10. Storm T (2010) KiKS is a Khepera simulator. http://www.tstorm.se./projects/kiks/
11. Berres M, Meyer KD (1988) $\lambda$- additive fuzzy measures on measure spaces. Fuzzy Sets and Systems. 27:159-169
12. Gilboa I, Schmeidler D (1994) Additive representations of non-additive measures and the Choquet integral. Ann Oper Res. 52(1):43-65