Synthesizing Adaptive Navigational Robot Behaviours Using a Hybrid Fuzzy A* Approach

Antony P. Gerdelan and Napoleon H. Reyes

Summary. Previously, we have devised a novel Hybrid Fuzzy A* algorithm that seamlessly integrates the forward planning feature of A* and the refined reactionary robot maneuvering capabilities of Fuzzy Logic in a real-time simulation environment. This paper further explores the uncharted domain of synthesizing three primary robot maneuvering behaviours, namely target pursuit, obstacle avoidance and opponent evasion in an adaptive compact Hybrid Fuzzy A* navigation system. In addition, this work sheds some light onto the dark pits of the previous Fuzzy A* architecture proposed, as the former Hybrid approach did not account for the necessity of evasive behavior, and so modifications to the forward planning layer are deemed to be necessary. In light of this, this chapter presents a new undesirability component that is injected into the A* algorithm, as well as optimisations to the cascade of fuzzy systems architecture that calculates the robot speed and angle adaptively. Empirical results are also presented that attest to the algorithm's robustness when faced with a formidable army of moving obstacles while in pursuit of a target, as well as evading multiple opponents.

Key words: Autonomous navigational systems, Path planning, Fuzzy logic, The A* algorithm, Robot soccer.

1 Introduction

The evasion algorithm presented here is designed to augment the path-planning layer of the previously developed Hybrid Fuzzy A* Robot Navigation System [1], and relies on receiving reliable threat information from the sensors of the robot system. After receiving information concerning threats, a series of maps (conceptual grids) is created to represent the robot's environment and a modified version of the A* algorithm is used to construct a shortest path to a goal, avoiding obstacles and evading potential threats to the robot. The system presented in this chapter has been created to extend navigation systems specifically for the game of robot soccer, but is applicable to problem domains

with the same key features. The problem domain for which this algorithm is intended consists of:

- A complex, three-dimensional environment
- Multiple hostile agents to evade
- A dynamic, moving target to pursue
- A static goal location
- Static and dynamic obstacles to avoid

The evasion algorithm is based on the premise that whilst seeking a target, a robot must also avoid obstacles and evade competing robots. The game of robot soccer operates in real time at very high speeds, and therefore demands extremely fast processing. The window of time available for calculation is approximately 33 ms and must be shared with machine vision processing and other system components. Navigation systems must be fast enough to remain synchronous with the state of the robots, and therefore a balance between speed of calculation and optimality must be struck.

2 General System Architecture (Fig. 1)

Environment information is collected by the sensors and analysed for key environment features. Environment features required for the planning layer with the evasion algorithm are:

- Hostile agent locations and headings
- Obstacle locations
- A goal location
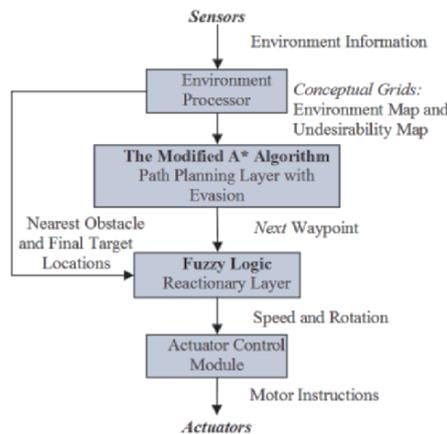
Fig. 1. Architecture of the augmented hybrid fuzzy A* system

These information are then used to create conceptual grids (maps) of the environment which are then passed to the planning layer. The environment processor must create two conceptual grids:

– An environment map
– An undesirability map

An environment map marks obstructed areas of the environment to be excluded from the search domain of the path-finding algorithm. An undesirability map awards undesirability ratings to all areas of the environment based on the level of perceived threat from hostile agents to that area. The planning layer draws on both maps to plan a path toward its goal location, avoiding obstacles and undesirable areas where possible. The waypoints of the path are passed down to the Fuzzy Logic Control layer, which refines robot movement and reacts to avoid immediate obstacles. Defuzzified outputs for speed and rotation of the robot are sent to the actuator control module, where speed and rotation outputs are decomposed into specific motor instructions.

3 The Evasion Algorithm

The evasion algorithm presented in this chapter is an extension of a dynamic A* path-finding algorithm [1]. Navigation systems employing dynamic A* path-finding operate on a 2D grid [2–4], representational of the real environment, and cells of the grid correspond to nodes in the search domain of the A* algorithm.

3.1 Dynamic A* Path Finding

For the A* path finding to operate in a dynamic environment, the conceptual grid must be continually regenerated, keeping up-to-date information on the locations of moving obstacles. This allows robots to recalculate paths on the fly when dynamic obstacles move and obstruct the intended path. Cells containing obstacles are excluded from the search process. Figure 2 illustrates a path calculated on such a grid, where obstructed nodes have been excluded from path calculation. Other nodes are awarded an f* score from the formula:

$$f^* = h^* + g^* \tag{1}$$

where h* represents a heuristic distance from the examined node to the goal node, and where g* represents the sum of distances between nodes; from the initial node to the examined node. The A* algorithm finds the shortest path by searching the domain of nodes and connecting nodes with the lowest f* scores from the start node to the goal. As can be viewed in Fig. 3, the h* values for nodes in 2D conceptual grid can be represented in three dimensions, where z-axis represents the h* value for each node.

Fig. 2. A Sample A* path finding result *(directed arrows)* vs. Hybrid Fuzzy A* Path Finding *(curve)*

Fig. 3. Example h* values for cells in a conceptual grid

Figure 3 illustrates the h* values for cells in a conceptual grid, where a start node is at grid position (1, S1) and a goal node at grid position (13, S13). We can clearly see that, excluding obstacles, the shortest path is in a straight line; from the highest point to the lowest.

Fig. 4. Undesirability values awarded to grid cells surrounding a hostile agent

3.2 Undesirability Maps

For the path-planning layer of the navigation system to evade opponents, it must be provided with information detailing the level of threat to areas of the environment. To this end, a second conceptual grid of the environment is created, and each cell is awarded a value based on the undesirability of the area. A function is required to determine the relative undesirability of each cell. The valuing function must be tailored to the specific problem domain and environment.

For the game of robot soccer levels of undesirability centre on each hostile agent; a teardrop-shaped field of undesirability extends away from the agent (refer to Fig. 4). Areas of highest undesirability are immediately in front of the agent. Areas beside the agent are less undesirable to reflect the reduced threat to those areas, as the agent must rotate before directly threatening those areas. Areas behind the agent are awarded a lower undesirability value also; reflecting the reduced threat of the agent reversing or turning around.

Figure 5 illustrates undesirability values in three dimensions for the same environment as illustrated in Fig. 3, where the z-axis represents the undesirability value of each cell. This effect can be compared to the implementation of navigation systems employing potential field methods [5–8], but extends the potential field concept by preempting the movements of hostile agents and their direction of travel.

3.3 An Evasive Path-Finding Algorithm

Undesirability values and heuristic distance values can be combined to form a new f* score, balancing the weight of heuristic distance with that of undesirability:

$$f^* = h^* + u^* + g^* \tag{2}$$
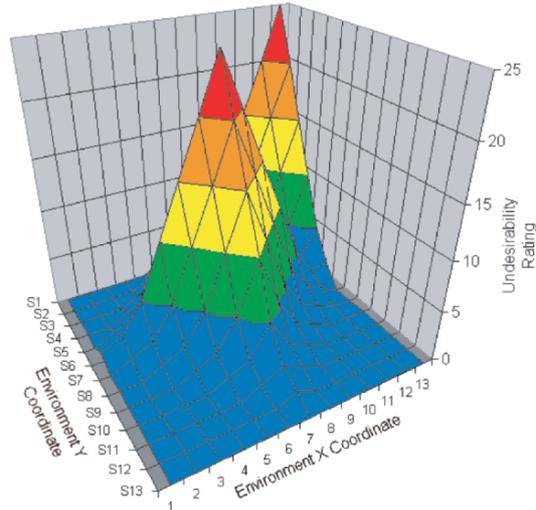
Fig. 5. Example undesirability values for cells in a conceptual grid

Equation (2) presents a new formula; undesirability has been factored into the heuristic component of the A* algorithm, where h* represents the heuristic distance from the examined node to the goal node, where u* represents the undesirability value of the examined node, and where g* represents the cumulative distance from the start node to the examined node.

The combined heuristic distance and undesirability values can be represented in three dimensions for a "hills-and-valleys" effect, where the z-axis represents the combined values for each cell in the conceptual grid. Figure 6 illustrates a combination of the heuristic distance values represented in Fig. 3 and the undesirability values represented in Fig. 5. We can see that, as the algorithm will try to construct a path with lower h* + u* values, it will no longer plan a path directly to the goal through the centre of the grid, but will circumnavigate the undesirable areas en route to the goal. The cumulative g* value ensures that a shortest possible path is created, and not simply the most "downhill" path.

3.4 Considerations for Evasion

Experimentation with simulation has shown that the weighting given to undesirability values is of great importance to the effectiveness of the evasion system. Values that are too low will effectively produce a system that has no evasive behavior undesirable regions will be outweighed by the path-finding components of the f* score formula. Referring to Fig. 6, undesirability values that are too low would be observed as the peaks created by u* disappearing below the sloping area created by h*. Experiments have shown that if
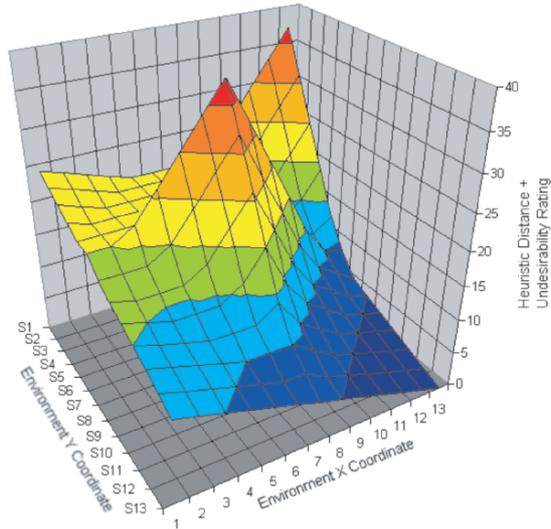
Fig. 6. h* + u* values for cells in a conceptual grid

undesirability values are too high, the undesirability component u* of the formula will overcome the path-finding components, compromising the reliability of the algorithm to produce consistent paths. A balance must be struck by choosing an appropriate range of undesirability values. Undesirability values that are too high would be observed by the total dominance of the u* peaks over the h* slope. The operation taken to determine the undesirability values for nodes where areas of threat overlap will result in subtle changes to robot behavior. If, in the problem domain of the robot, areas in between multiple threats are even more undesirable than areas near one threat alone, it may make sense to award the area affected the sum of all of the overlapping undesirability values. Otherwise, the maximum of the values may be the most effective choice.In the game of robot soccer, robots attempting to evade other robots are often crushed against the walls. In this problem domain, the undesirability of areas near static obstacles (the walls) can be increased to good effect. For broader applicability, areas where a robot has more room to evade to maneuver and change course should be less undesirable than those areas where a robot is more confined. The direction, speed and other details of hostile agents can be incorporated into the undesirability rating function, so that a very accurate representation of threat can be created in the threat map. Areas further in front of fast-moving hostile agents are awarded high undesirability ratings, whilst ratings for areas to the sides and behind those agents are reduced, as it takes those agents longer to turn around. This additional information makes more accurate threat maps, but requires additional calculation and is of diminishing importance to the robot as hostile agents are further away from it.

4 Cascade of Fuzzy Systems

Robot navigation systems employing Fuzzy Logic [9–14] require input information about a robot's target; the distance between the robot and the target, and the angle between the robot's heading and the target. Utilising the same inputs, the proposed Fuzzy system architecture embodies multiple fuzzy systems that collectively perform angle and speed refinements for the tasks of target pursuit and obstacle avoidance. As can be seen in the diagram of the Reactionary Layer (Fig. 7), the Path Planning Layer feeds the next intermediary waypoint to a cascade of Fuzzy systems which paves the way for a smooth robot maneuvering towards the target.

Such refined robot movements are made precisely to suit the prevailing circumstances. Target pursuit is carried out by the system whenever it is safe to do so, and obstacle avoidance is instantaneously engaged with the onset of any opponent interference. On the top layer is the Fuzzy system for target pursuit (Fuzzy System 1) that reacts on two inputs, namely the robot's distance from the target, and the difference between the robot's heading angle and target. The main task of such system is to calculate the correct turning
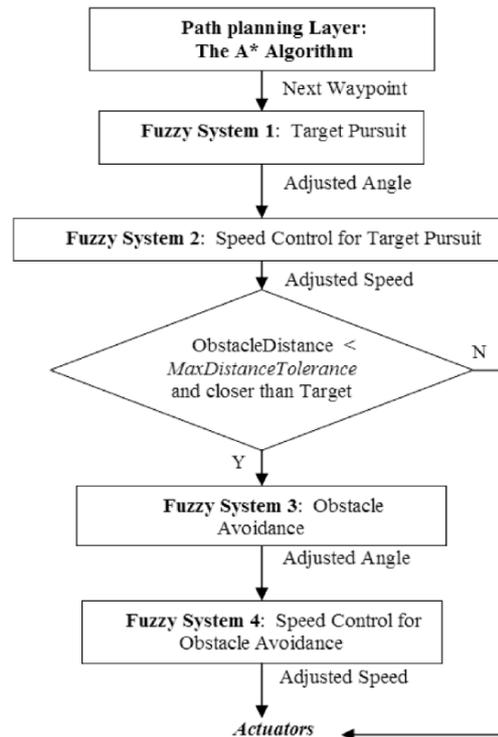
Fig. 7. Reactionary layer: cascade of fuzzy systems

Table 1. Fuzzy associative memory matrix for angle control: target pursuit

	NEAR	FAR	VERY FAR
SMALL	Mild Turn	Mild Turn	Zero Turn
MEDIUM	Med Turn	Mild Turn	Mild Turn
LARGE	Sharp Turn	Med Turn	Med Turn
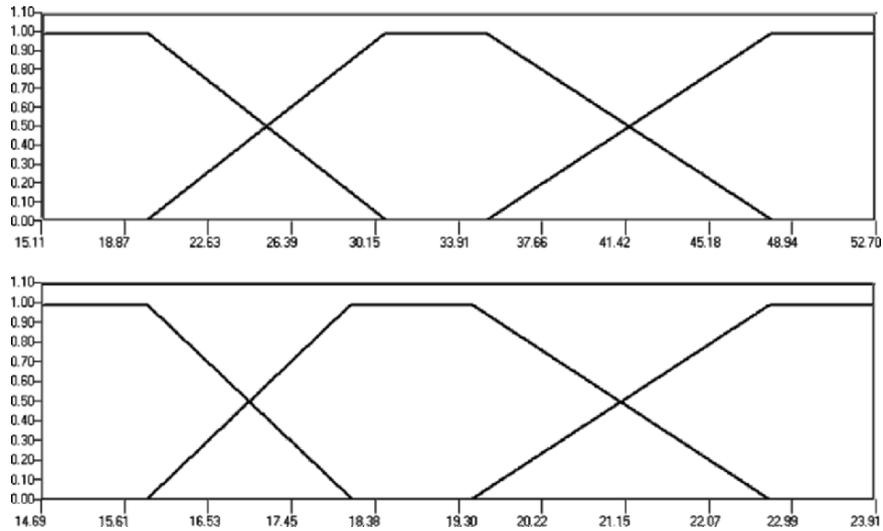
Fig. 8. Fuzzy membership functions for speed control: target pursuit – angle *(top)* and Distance *(bottom)*

angle towards the target relative to its current orientation. Table 1 depicts the collection of rules that dictate the correct turning angle for a combination of distance and angle conditions. Such fuzzy sets are defined using trapezoidal membership functions (Fig. 8). As an example, one of the rules states that:

> If the **Distance** from the Target is *NEAR* and the **Angle** from the Target is *SMALL* Then the robot should make a **Mild Turn**.

4.1 Taking Advantage of Angle Symmetry

It is worth mentioning that the design for the fuzzy associative memory matrix takes advantage of the angle symmetry; thereby, considering only the right-half of the angles involved, from [0, 90] and [270, 360]. As can be viewed in Fig. 9, the angles were partitioned only into three overlapping parts, each with its own corresponding fuzzy set. Using this simplified approach, the size of the FAMM was considerably reduced, since both left and right cases were
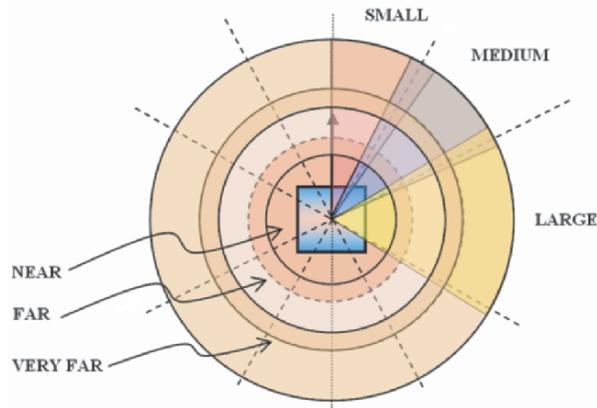
Fig. 9. Fuzzy sets for angle and distance

Table 2. Fuzzy associative memory matrix for speed control: target pursuit

	NEAR	FAR	VERY FAR
SMALL	Med Speed	Fast Speed	Very Fast
MEDIUM	Slow Speed	Med Speed	Fast Speed
LARGE	Very Slow Speed	Slow Speed	Slow Speed

accounted for using the same generic FAMM. In particular, for cases where the obstacle or target is found on the left-hand side, considering angles from [90, 270] counter-clockwise, relative to the robot's orientation, the fuzzy output is simply negated. Moreover, the Fuzzy systems were designed to respond by taking the minimum turning angle towards the desired robot orientation. This reactionary robot pursuit movement is further enhanced by yet another Fuzzy system that handles speed control based on the same inputs fed into Fuzzy System 1. As an example, a fuzzy rule for speed control comes in the following form: If the Distance from the Target is VERY FAR and the Angle from the Target is SMALL Then the robot should move Very Fast. Finally, the two Fuzzy Systems at the bottom were designed to perform course corrections to account for cases where obstacles are close to the robot. Similar to Fuzzy System 1, except that it is considering an obstacle instead of the target, Fuzzy System 3 adjusts the robot's steering angle based on the robot's distance from the obstacle, and difference between the robot's heading angle and the angle to the obstacle. In conjunction with Fuzzy System 3, Fuzzy System 4 deals with speed adjustment, which is also similar to Fuzzy System 2 (Table 2), except that its rule base is designed to avoid collisions.

4.2 Limits of the Reactionary Fuzzy Systems

Despite the system's ability to perform refined course corrections to pursue the target and avoid the obstacles, there are cases however where forward planning is necessary to prevent the robot from taking routes that could lead to it getting trapped. Since the fuzzy systems do not take into account the directions of the moving obstacles, using the fuzzy system solely is not enough to prevent collisions completely. Thus, the A* algorithm is used to guide the cascade of fuzzy systems.

5 Conclusions

This chapter has extended our previous paper on a novel Hybrid Fuzzy A* navigational system, inculcating a predictive quality into autonomous robot navigation. Robot path planning is now able to second guess the movements of hostile agents in order to evade the onslaught of potential threats in real time.

References

1. Gerdelan, A.P., Reyes, N.H.: A novel hybrid fuzzy a* robot navigation system for target pursuit and obstacle avoidance. In: Proceedings of the First Korean–New Zealand Joint Workshop on Advance of Computational Intelligence Methods and Applications, Volume 1 (2006) 75–79
2. Yap, P.: Grid-based path-finding. In: Advances in Artificial Intelligence: 15th Conference of the Canadian Society for Computational Studies of Intelligence, AI, Volume 2338, Calgary, Canada (2002) 44
3. Reynolds, C.: Steering behaviors for autonomous characters. In: Proceedings of Game Developers Conference (1999)
4. Botea, A., Muller, M., Schaeffer, J.: Near optimal hierarchical path-finding. Journal of Game Development **1**(1) (2004) 7–28
5. Newman, W., Hogan, N.: High speed robot control and obstacle avoidance using dynamic potential functions. In: Proceedings of the 1987 IEEE International Conference on Robotics and Automation, Raleigh, North Carolina (1987) 14–24
6. Koren, Y., Borenstein, J.: Critical analysis of potential field methods for mobile robot obstacle avoidance. Submitted for publication in the IEEE Journal of Robotics and Automation (1990)
7. Koren, Y., Borenstein, J.: Potential field methods and their inherent limitations for mobile robot navigation. In: Proceedings of IEEE Conference on Robotics and Automation, Sacramento, CA (1991) 1398–1404
8. Ge, S.S., Cui, Y.J.: New potential functions for mobile robot path planning. IEEE Transactions on Robotics and Automation **16** (2000) 615–620
9. Bonarini, A., Invernizzi, G., Labella, T., Matteucci, M.: An architecture to coordinate fuzzy behaviors to control an autonomous robot. Fuzzy Sets and Systems **134**(101) (2003)

10. Meng, Q., Zhuang, X., Zhou, C., Xiong, J., Wang, Y., Wang, T., Yin, B.: Game strategy based on fuzzy logic for soccer robots. In: Proceedings of the 2000 IEEE International Conference on Systems, Man, and Cybernetics, Volume 5 (2000) 3758–3763
11. Oller, A., de la Rosa, J.L., Garca, R., Ramon, J., Figueras, A.: Micro-robots playing soccer games: a real implementation based on a multi-agent decision-making structure. International Journal of Intelligent Automation and Soft Computing
12. Wong, C.C., Chou, N.W., Hwang, C.P., Tsai, C.H., Shyu, S.R.: A method for obstacle avoidance and shooting action of the robot soccer. In: Proceedings of the 2001 IEEE International Conference on Robotics and Automation (2001) 3778–3782
13. Hwang, C.L., Chang, N.W., Han, S.Y.: A fuzzy decentralized sliding-mode control for car-like mobile robots in a distributed sensor-network space. In: Proceedings of the Third International Conference on Computational Intelligence, Robotics and Autonomous Systems (2005)
14. Baltes, J., Hildreth, N., Lin, Y.M.: The all botz robocup team. In: Proceedings of the PRICAI Workshop on RoboCup, Singapore (1998)