

## Responsive Motion Control for Robot Soccer Navigation Using Adaptive Social Force Framework

Bima Sena Bayu Dewantara<sup>1</sup>, Bagus Nugraha Deby Ariyadi<sup>2</sup>, Hary Oktavianto<sup>3</sup>

<sup>1,2</sup>Department of Informatics and Computer Engineering, Politeknik Elektronika Negeri Surabaya, Indonesia

<sup>3</sup>Department of Electrical Engineering, Politeknik Elektronika Negeri Surabaya, Indonesia

### Article Info

#### Article history:

Received Nov 17, 2020

Revised Jul 28, 2021

Accepted Aug 13, 2021

#### Keywords:

Navigation,  
Soccer robot,  
Social force model,  
Fuzzy inference system,  
Simulation

### ABSTRACT

This paper presents a modified Social Force Model (SFM) for navigation control of a soccer robot application. We modified the way of determining the parameter value of the gain factor,  $k$ , of the SFM using the Fuzzy Inference System (FIS), so that the value of the gain factor,  $k$ , is adaptive. The purpose of the gain factor adaptation is that the robot can move responsively but not over-reactive when it encounters an obstacle at high speed, which is a weakness of SFM with fixed parameters. Modification of SFM parameters using FIS is hereinafter referred to as the Fuzzy-based Social Force Model (F-SFM). We used this technique on a soccer robot with an omnidirectional drive platform with three motors. As experiments, several modifications to the FIS rules were made and compared to the SFM with fixed parameters. The simulation-based experimental results show that the proposed method outperforms the SFM method with fixed-parameters, and the computation time does not differ significantly so that it can be applied for real implementation.

Copyright © 2021 Institute of Advanced Engineering and Science.  
All rights reserved.

### Corresponding Author:

Bima Sena Bayu Dewantara,  
Department of Informatics and Computer Engineering,  
Politeknik Elektronika Negeri Surabaya,  
Raya ITS, Keputih, Sukolilo, Surabaya, Indonesia.  
Email: bima@pens.ac.id

### Nomenclature

$V_x$  = the robot's velocity in x-direction

$V_y$  = the robot's velocity in y-direction

$V_\theta$  = the robot's angular velocity to control its heading

$v_x^i$  = the speed of i-th motor in the x-axis direction of the wheel rotation

$F^n$  = the navigation force

$F^g$  = the attractive force towards the desired goal

$F_j^o$  = the repulsive force concerning obstacle j-th

$k$  = the gain factor of the repulsive force,  $F_j^o$

$r_{ij}$  = the sum of the robot's i-th radius and the obstacle's j-th radius

$d = d_{obs} = d_{ij}$  = the relative distance between the robot i-th and the obstacle j-th

$\sigma$  = the effective range within which the system will begin to respond to the stimuli

$f_s^o$  = the social force caused by the closest obstacle

$f_p^o$  = the physical force caused by the closest obstacle

$m$  = the robot's mass

$\tau$  = the amount of time the system needs to update the robot's speed,  $v_t$

$v_o$  = the maximum speed of the robot

$v_t$  = the present speed of the robot

$a$  = the difference between  $v_o$  and  $v_t$

$\alpha$  = the direction of the navigation force,  $F^n$

$\beta$  = the direction of the target location relative to the robot

$\theta$  = the direction of the actual robot heading

$\gamma = \vec{e}_{ij}$  = the direction of the obstacle force

## 1. INTRODUCTION

Over the most recent ten years, robot soccer competition has become a stage for the improvement of different sorts of innovation to answer the undeniably significant level of challenges. One of the individual capacities of a soccer robot that is still fascinating and challenging to develop is the robot's capacity to navigate at high speed. Navigating at high speed greatly impacts the responsiveness of the robot when it encounters an obstacle. Controlling the robot at high speed requires a reliable and fast control system.

A few past investigations in the field of mobile robot soccer have utilized a lot of potential field methods and their modifications to plan robot navigations. The potential field method is widely used even though there are weaknesses in that it is often trapped in local optima [1]. Improvements were made by adding several techniques such as Simulated Annealing [2] and evolutionary algorithm like Genetic Algorithm (GA) [3]. However, the use of GA for optimizing the path is time consuming. Navigation problems are also often solved with the path and motion planning approach. Albab et.al. [4] used Genetic Algorithm (GA) to choose the best path for robot movements. While Afakh et.al. [5] utilized a pre-defined path to guide the robot's movement. Rodriguez et.al. [6] and Rahman et.al. [7] used the Rapidly-Exploring Random Tree (RRT) to find a path between the position of the robot and the target goal. RRT is very effective, but it will take a long time to find the best path for the robot. Abiyev et.al. [8] also used a tree combined with Fuzzy Logic Control (FLC) to control robot navigation. However, the processing time is still quite high at 0.9 seconds. Renardi et.al. [9] proposed the obstacle avoidance using subtargets. This method looks quite effective and good for avoiding dynamic obstacles, however, unfortunately, it is still tested at low robot speeds (below 1 m/sec).

The social force model (SFM) [10] is one of the many popular local motion planning methods that commonly used to investigate an agent motion and interaction in a small scale social environment. It uses the concept of calculating the resultant navigational force of the obstacle repulsion force and the destination attraction force. When it comes to robot soccer, the movement of a robot is very much determined by certain goals and purposes, such as avoiding an obstacle [11], pursuing the ball [12] or dribbling. When navigating, the behavior of soccer robots is very much different from robots for other social-context based applications, for example guiding [13], following a target person [14], accompanying [15], and so on. The responsiveness is the main key for the robot to successfully finish its task. It is determined by how swiftly the robot moves to respond to the stimulus it receives, i.e. the distance and orientation of the obstacle. Therefore, the robot's ability to adjust its control parameters will affect how much response will be generated. Two parameters,  $k$  and  $\sigma$ , are commonly predefined in the SFM framework. The level of responsiveness of the robot in mobile robot soccer is substantially governed by the value of  $k$ .

Research related to determining the value of SFM parameters has been reported by previous researchers. The use of fixed SFM parameters strategies were demonstrated by Ferrer [15], Zanlungo [16], and Luber [17]. The SFM parameters were globally optimized by using the Genetic Algorithm (GA) [18]. The main demerits of the global optimization is that their approach cannot be generally implemented in different space conditions, for example, indoor and outdoor, wide and narrow spaces. Dewantara [19] proposed a strategy for adaptively changing the SFM parameter values by using Reinforcement Learning (RL) [20]. It was solved the generalization problem, however, the RL state-action pair was still determined manually and the sensor reading area needs to be discretized to reduce the number of states. This makes the robot responses using same action for a little bit of changing in sensor readings. Other research on SFM modification was conducted by Sent *et.al.* [21] which involved Fuzzy Logic Control (FLC) to modify pedestrian behavior to make it better interpretable. The FLC modifies several types of forces by involving as many as 22 Fuzzy rules. The proposed method was successful enough to be tested using a pedestrian simulation, but the time required to run the FLC when modifying the SFM output increased by 10% to 50% compared to using SFM alone. Besides that, the difficult tuning process has to be done with several revisions to get the FLC results that are assumed to be the best.

We introduced the Fuzzy-based Social Force Model (F-SFM) in this study, in which the Fuzzy Inference System (FIS) [22] is used to determine the value of  $k$  using the pair of input stimulus,  $d$  and  $\gamma$ , between the obstacle and the robot. Both inputs will be processed using FIS to produce an appropriate value of  $k$ , which is suitable so that the level of responsiveness of the robot matches the stimulus received. In this study, we show the following contributions:

- This strategy provides a quick way to design a robot soccer's level of responsiveness.
- The determination of the gain factor,  $k$ , is easy and always proportional to all of possible distances and directions of the obstacle relative to the robot.

## 2. RESEARCH METHOD

### 2.1. Omnidirectional drive mobile robot

The mobile robot soccer model we use is a three-wheeled omnidirectional drive platform from Festo, called as Robotino [23]. Figure 1 shows a configuration of our robot model. Referred to Figure 1, the kinematic and inverse kinematic formulations of the robot are expressed as follows:

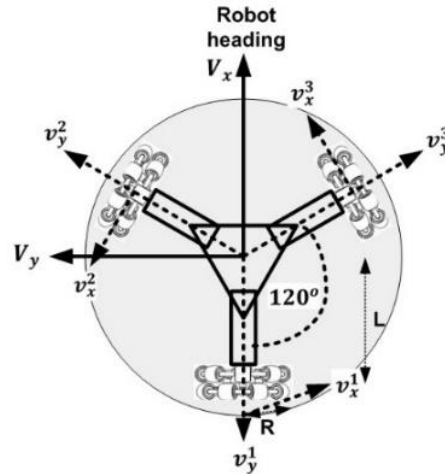


Figure 1. The three-wheels layout of the omnidirectional drive mobile robot platform

$$\begin{bmatrix} V_x \\ V_y \\ V_\theta \end{bmatrix} = \begin{bmatrix} 0 & -\frac{R}{\sqrt{3}} & \frac{R}{\sqrt{3}} \\ -\frac{R}{3} & -\frac{2R}{3} & \frac{2R}{3} \\ \frac{R}{3L} & \frac{R}{3L} & \frac{R}{3L} \end{bmatrix} \begin{bmatrix} v_x^1 \\ v_x^2 \\ v_x^3 \end{bmatrix}, \quad (1)$$

$$\begin{bmatrix} v_x^1 \\ v_x^2 \\ v_x^3 \end{bmatrix} = \begin{bmatrix} 0 & -\frac{1}{R} & \frac{L}{R} \\ -\frac{\sqrt{3}}{2R} & \frac{1}{2R} & \frac{L}{R} \\ \frac{\sqrt{3}}{2R} & \frac{1}{2R} & \frac{L}{R} \end{bmatrix} \begin{bmatrix} V_x \\ V_y \\ V_\theta \end{bmatrix}, \quad (2)$$

The speeds of wheel  $i$ -th in the  $x$ -axis and  $y$ -axis of wheel  $i$ -th workspace are  $v_x^i$  and  $v_y^i$ , respectively. However, only the speed of each motor in the  $x$ -axis direction of the wheel rotation,  $v_x^i$ , is used in kinematic and inverse kinematic calculations.

### 2.2. Fuzzy-Social Force Model

#### 2.2.1. Social Force Model

Social Force Model (SFM) [10] was widely used to model how people interact each other in the case of pedestrian dynamics by utilizing the concept of attractive and repulsive forces. When applied to a robot to imitate the human-human interaction, at that point, the robot navigates by taking into account the idea of SFM as shown in Figure 2 and is expressed in Eq. 3. Eq. 4 depicts the formulation of  $F^g$ .

$$F^n = F^g + \sum_{j=0}^M F_j^o, \quad (3)$$

$$F^g = m \cdot a = \frac{m \cdot d(v_0 - v_t)}{\tau}, \quad (4)$$

From Figure 2, the goal,  $G(x, y)$ , can be the position of the balls or goals on the field. The obstacle,  $O(x, y)$ , is the position of the obstacle or opposing robot.  $R(x, y)$  is the current position of our robot. Then, the obstacles themselves can be static or dynamic. Eq. 5 – Eq. 7 are formulas for the forces created by obstacles. The specification of the SFM parameters are shown in Table 1.

$$F^o = f_s^o + f_p^o, \quad (5)$$

$$f_s^o = k \cdot \exp\left(\frac{r_{ij} - d_{ij}}{\sigma}\right) \vec{e}_{ij}, \quad (6)$$

$$f_p^o = k \cdot (r_{ij} - d_{ij}) \vec{e}_{ij}, \quad (7)$$

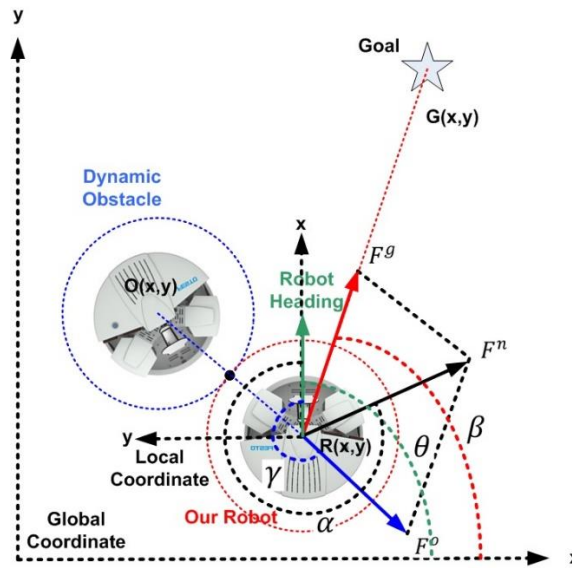


Figure 2. Forces that work within the SFM framework and their causes. The star sign is the goal position. The circle around the robot is the proxemic distance that is used as a limit on when the SFM should respond.

Table 1. The SFM parameters

Parameters	Value
Robot mass, $m$	20 kg
Updating time, $\tau$	0.1 sec
Max. speed, $v_0$	2 m/sec
Proxemic type	circle
Proxemic radius, $r$	0.5 m
Effective range, $\sigma$	0.5 m

### 2.2.2. Fuzzy Inference System

The Fuzzy Inference System (FIS) is utilized to respond proportionally to the obstacle's stimuli,  $d$  and  $\gamma$ , and to provide an effective gain factor,  $k$ , of the obstacle's forces. Fuzzification and membership function, inference engine and rule base, and defuzzification are the three steps in the FIS process. Because a proportional ratio between the input and output pair is required, we selected a triangular shape for the membership functions. The following formula is used to calculate the degree of the triangular shape-based membership function.

$$\mu = \begin{cases} \frac{x-u}{v-u} & \text{if } u < x \leq v \\ \frac{w-x}{w-v} & \text{if } v < x \leq w \\ 0 & \text{if others} \end{cases} \tag{8}$$

Where  $\mu$  is the membership function's degree,  $x$  is the crisp input,  $u$  is the membership function's left boundary,  $v$  is the membership function's center, and  $w$  is the membership function's right boundary. Figure 3 shows how each input's membership function is handled. To follow Hall's proposed proxemic distance [24], the relative distance is separated into four segments. We also divided the relative direction into four segments to make the calculation easier.

From Figure 3, we have two different forms of memberships,  $\mu^d$  and  $\mu^\gamma$ , which reflect the degree of membership function of the relative distance,  $d$ , and the relative angle,  $\gamma$ , between the robot and the obstacles, respectively. The inference value can be calculated using the following formula.

$$\mathbf{B} = [\mu_m^d \mu_m^\gamma]_{m=1,2,\dots,M} \tag{9}$$

where  $\mathbf{B}$  is the T-norms between  $\mu_m^d$  and  $\mu_m^\gamma$ . There are  $M$ -memberships for each relative distance and relative direction, giving us a total of  $M^2$  dimensions for which a Fuzzy rule  $R$  must be defined. Here, we define  $R =$

{VH, H, H, H; H, H, M, L; M, M, L, L; L, L, L, VL}, where VH means Very high Risk, H means High Risk, M means Medium, L means Low Risk, and VL means Very Low Risk. Figure 4 shows the design of our Fuzzy's rule.

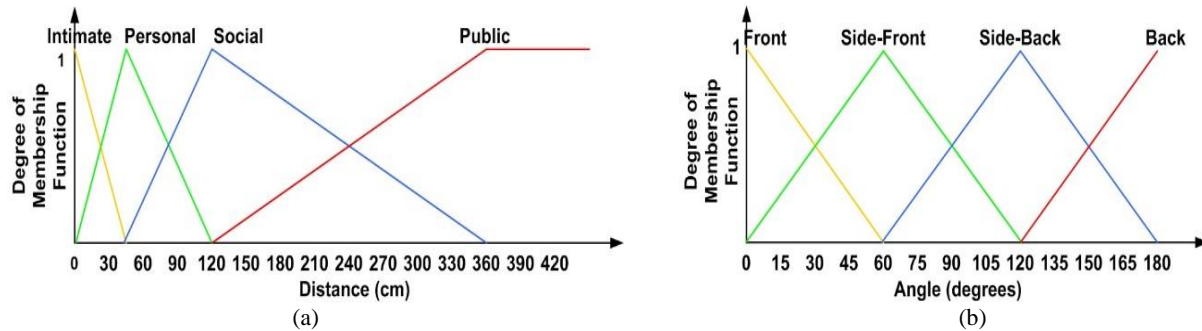


Figure 3. The Fuzzy Membership Functions (MF) using triangular shape type. (a) MF of  $d$  and (b) MF of  $\gamma$ .

R		Distance $d$			
		Intimate	Personal	Social	Public
Direction $\gamma$	Front	VH	H	H	H
	Side-Front	H	H	M	L
	Side-Back	M	M	L	L
	Back	L	L	L	VL

Figure 4. Fuzzy's rule

Finally, defuzzification is accomplished by turning the inference engine's fuzzy output to a crisp using the Center of Area (CoA) method as follows.

$$k = \frac{\sum_{n=1}^N B_n R_n}{\sum_{n=1}^N B_n}, \tag{10}$$

where  $k$  is the expected gain value, and  $n = 1, 2, \dots, N$  is the number of inferences  $B$  as well as the number of  $R$ .

### 2.3. System design

Our proposed system design is presented in Figure 5. The FIS is utilized to reasoning the correlation between the pair of  $d$  and  $\gamma$ , of the obstacle to the robot with a proportional value of  $k$ . The  $k$ -value,  $d$ ,  $\gamma$ , and  $\beta$ , were then fed into the SFM. The combination of all parameters will result in a pair of driving speeds,  $v_t$ , and its direction,  $\alpha$ . Figure 6 illustrates a schematic of a match when all components are on the soccer field. Parameter measurements that apply to each component can be calculated by the following formula.

$$d_{goal} = \sqrt{(G_x - R_x)^2 + (G_y - R_y)^2} \tag{11}$$

$$d = d_{obs} = \sqrt{(O_x - R_x)^2 + (O_y - R_y)^2} \tag{12}$$

$$\beta = \tan^{-1} \left( \frac{G_y - R_y}{G_x - R_x} \right) \tag{13}$$

$$\gamma = \tan^{-1} \left( \frac{O_y - R_y}{O_x - R_x} \right) \tag{14}$$

### 2.4. Controlling robot heading direction

It's not easy to control the heading direction of an omnidirectional based mobile robot. Comparing the direction of the actual robot heading,  $\theta$ , with the direction of the target location relative to the robot,  $\beta$ , is one way of controlling the robot heading. The following are the equations related to the orientation of the goal location for regulating the robot heading error.

$$err(t) = \beta(t) - \theta(t), \tag{15}$$

$$V_{\theta}(t) = k_p \cdot err(t) + k_i \cdot \sum err(t) + k_d \cdot (err(t) - err(t - 1)), \tag{16}$$

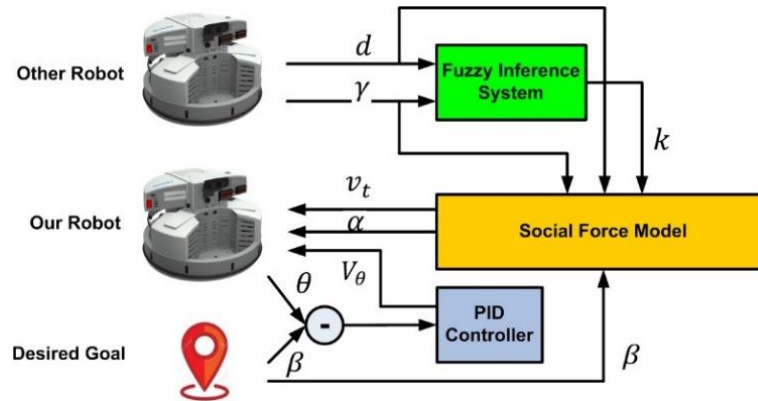


Figure 5. Our proposed system design

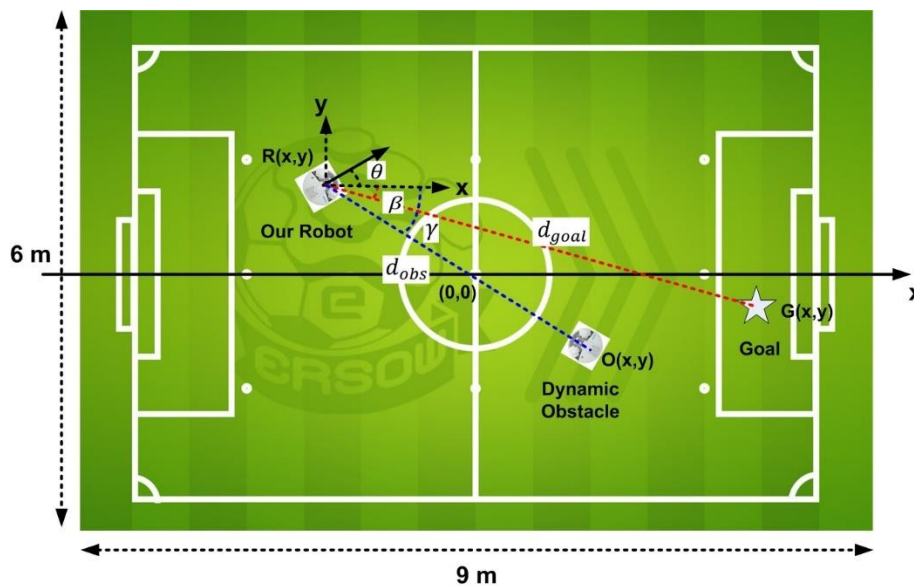


Figure 6. Illustration of a competition scheme where all components are on a soccer field.

where  $err(t)$  is the difference between  $\theta$  and  $\beta$ , at time  $t$ . In real conditions,  $\theta$  is obtained from the Inertial Measurement Unit (IMU) sensor readings. However, in this simulation, the value of  $\theta$  is obtained directly from the V-Rep application. The rotating speed of the robot is  $V_{\theta}(t)$ , and the proportional, integral, and derivative parameters of the PID controller are  $k_p$ ,  $k_i$ , and  $k_d$ , respectively. In this experiment, the values of  $k_p = 20$ ,  $k_i = 0.05$ , and  $k_d = 50$  are set by trial-and-error technique.

**2.5. The relationship between Social Force Model and mobile robot kinematics**

A mobile robot with an omnidirectional drive platform, as discussed in section 2.1, requires three motor speed outputs:  $v_x^1$ ,  $v_x^2$ , and  $v_x^3$ . The three outputs are the outcomes of calculating three inputs, namely  $V_x$ ,  $V_y$ , and  $V_{\theta}$ , according to the robot's inverse kinematics formulation. The following is how we express the relationship between SFM and the robot's inverse kinematic.

$$v_t = \frac{F^n \cdot t}{m}, \tag{17}$$

$$\alpha = \tan^{-1} \frac{F^{\theta} \sin(\beta) + F^o \sin(\gamma)}{F^{\theta} \cos(\beta) + F^o \cos(\gamma)}, \tag{18}$$

$$V_x = v_t \cdot \cos(\alpha), \tag{19}$$

$$V_y = v_t \cdot \sin(\alpha), \tag{20}$$

where  $\tau$  is set to 0.1 sec. Meanwhile, using the PID formulation in section 2.4,  $V_{\theta}$  is determined separately.

### 3. RESULTS AND DISCUSSION

#### 3.1. Experimental setup

The experiments were carried out in a simulation on a computer with the specifications shown in Table 2. These experiments focused on the response of the individual movement control of the soccer robot when pursuing the ball by avoiding obstacles that represent the opponent's robot. The scenario of the opponent's robot movement is divided into three, namely standing still, moving to cut off the movement of our robot, and moving to pursue the ball. We also compared the performance of F-SFM with other methods such as classical SFM and Artificial Potential Field (APF) [1]. The goal of these experiments is to see how responsive and safe the robot is at avoiding obstacles and achieving the goals. The number of iterations required by the robot when navigating to escape the opponent robot reflects its responsiveness. The fewer the iterations, the more precise the robot can avoid and the less control it requires. While the robot's movement is said to be safe when the robot does not collide the opponent's robot. The fewer collisions with the opposing robot, the better and safe the control the robot does.

Table 2. The specifications of hardware and software used

Hardware/Software	Specifications
CPU	Intel Core i7-8550U @1.8 GHz
RAM	16 GB
Graphics/GPU	NVidia GeForce 940MX
Operating System	Windows 10 Home 64-bit
Developer	Microsoft Visual C++ 2010 Express
Library	OpenCV 2.4.9
Simulator	V-Rep 3.6.2 [25]

#### 3.2. Responsiveness and safety

In this section, we investigated the robot responsiveness while avoiding static and dynamic obstacles using SFM with fixed  $k$ -value, APF with fixed  $\zeta$ -value, and SFM with adaptive  $k$ -value methods. In this experiment, we evaluated the responsiveness of the robot by varying the value of the constant gain factor,  $k$ , for the SFM with fixed  $k$ -value method,  $\zeta$ , for the APF with fixed  $\zeta$ -value method and varying the value of the Fuzzy output sets for the adaptive  $k$ -value method as shown in Table 3. Each experiment was repeated 10 times.

Table 3. The experimental schemes for fixed  $k$ -value and adaptive  $k$ -value

SFM		APF [1]			F-SFM (ours)					
Scheme	$k$ -values	Scheme	$\zeta$ -values	Scheme	Fuzzy output sets					
Fixed $k$ -value	#1 #2	Fixed $\zeta$ -value	#3 #4	Adaptive $k$ -value	#5 #6	VH	H	M	L	VL
	100 300		100 300		200 600	150	100	50	25	75

##### 3.2.1. Avoiding static obstacles

We examined the performance of the robot using the SFM framework to move from the starting position to the target position with the task of avoiding two static obstacles between the start and target positions. Experiments were carried out using APF with fixed  $\zeta$ -value, SFM with fixed  $k$ -value and SFM with adaptive  $k$ -value methods. The results of the comparison of the three methods are shown in Figure 7. The experimental results for each scheme are presented in Table 4.

From the Figure 7(a), we show the soccer robot equipped with APF with fixed  $\zeta$ -value attempted to avoid two static obstacles between the start position and the target position. The robot's responsiveness is represented by the number of iterations of 80 in average that needed by the robot to reach the target position for two schemes (#3 and #4). The robot demonstrated safely navigations by achieving only 55% of success rate. On the other hand, Figure 7(b) shows an experiment using SFM with fixed  $k$ -value, where the average of free collision or safe navigation is 18 out of 20 trials. This means that the robot successfully avoids the collision about 90%. This method also obtains lower number of iterations which means that the robot moves more responsive compared to the APF. Figure 7(c) shows the robot performances to avoid two static obstacles by using SFM with adaptive  $k$ -

Table 4. The experimental schemes for static obstacles

Scheme	Avg. Num. of Iterations	Avg. All	Static Obstacles	
			Num. of Free Collision	Percentage of safe navigation
#1	61		8 of 10	18 of 20
#2	67	64	10 of 10	90 %
#3	84		3 of 10	11 of 20
#4	76	80	8 of 10	55 %
#5	63		10 of 10	20 of 20
#6	57	60	10 of 10	100 %



value based on FIS. The robot can reach the goal by achieving 60 iterations in average for scheme #5 and #6. By using adaptive  $k$ -value, the average of free collision or safe navigation is 20 out of 20 trials. This means that the robot successfully avoids the collision about 100%.

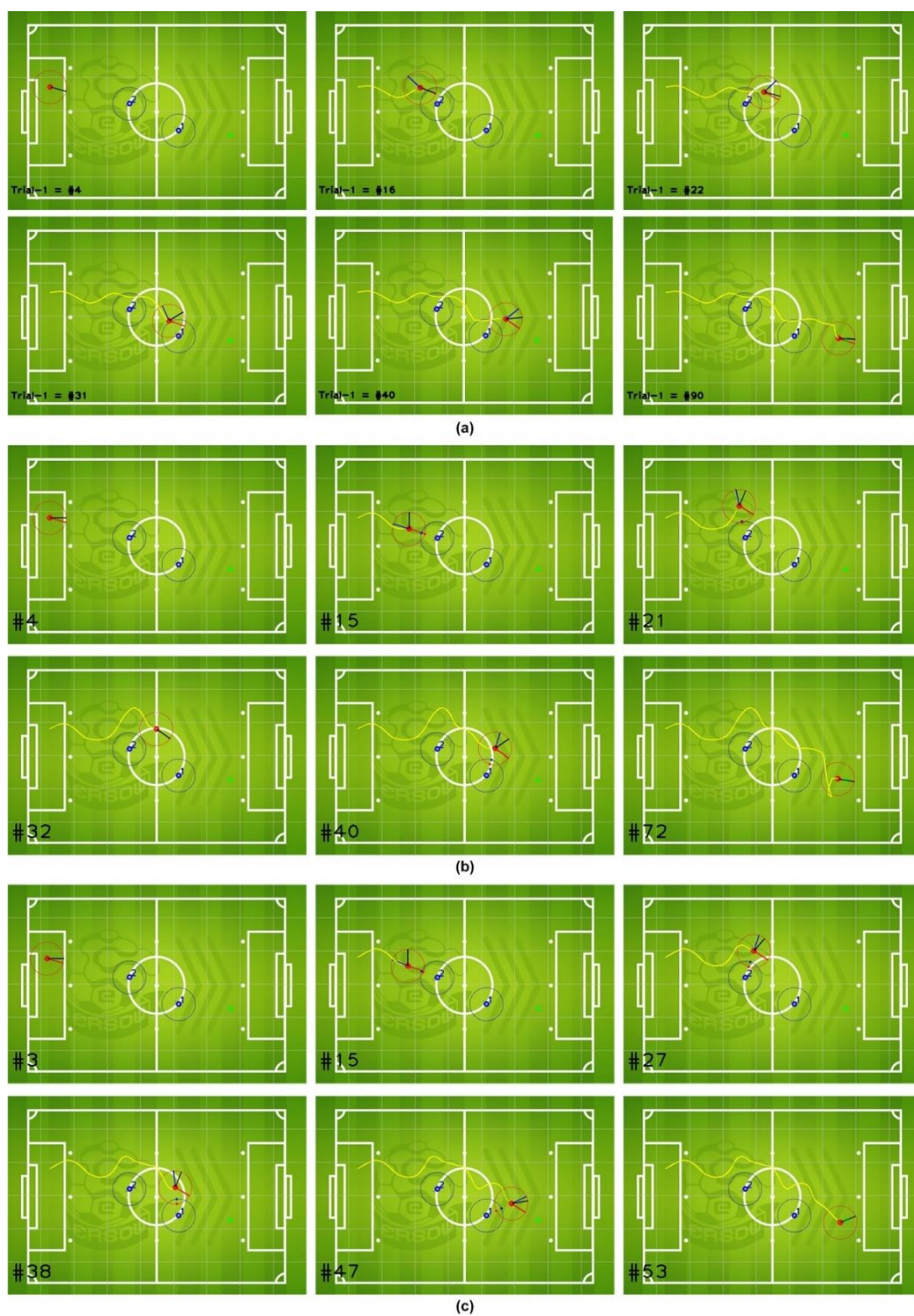


Figure 7. Examples of the robot's responsiveness when avoiding static obstacles. (a) using APF with fixed  $\zeta$ -value for scheme 3. (b) using SFM with fixed  $k$ -value for scheme 1. (c) using SFM with adaptive  $k$ -value for scheme 5. The yellow curve is the trajectory made by our robot



### 3.2.2. Avoiding dynamic obstacles

We examined the performance of the robot using the SFM framework to move from the starting position to the target position with the task of avoiding two dynamic obstacles between the start and target positions. We deliberately made two schemes, namely the dynamic obstacles movement cut our robot movement from two sides, namely left and right and the dynamic obstacles disturb our robot when pursuing the target position. The speed of the two dynamic obstacles is set at 1 m/s. Experiments were carried out using APF with fixed  $\zeta$ -value, SFM with fixed  $k$ -value and SFM with adaptive  $k$ -value methods. The results of the comparison of the experiments are shown in Figure 8 and Figure 9. The experimental results are shown in Table 5 and Table 6.

Table 5. The experimental schemes for dynamic obstacles where the opponent robots block our robot movement

Scheme	Avg. Num. of Iterations	Dynamic Obstacles		
		Avg. All	Num. of Free Collision	Percentage of safe navigation
#1	52	49	7 of 10	17 of 20
#2	46		10 of 10	85 %
#3	61	64	1 of 10	5 of 20
#4	67		4 of 10	25 %
#5	46	47	8 of 10	18 of 20
#6	48		10 of 10	90 %

From Tabel 5 and Figure 8(b), we show the robot tried to avoid two dynamic obstacles between the start position and the target position by performing obstacle avoidance abilities for different fixed  $k$ -values. The robot can reach the target by achieving 49 iterations in average for two schemes (#1 and #2). By using fixed  $k$ -value, the average of free collision or safe navigation is 17 out of 20 trials. This means that the robot successfully avoids the collision about 85%. Meanwhile, Figure 8(a) shows the robot performances to avoid two dynamic obstacles by using APF with fixed  $\zeta$ -value. The robot can reach the target by achieving 64 iterations in average for two schemes (#3 and #4). By using APF with fixed  $\zeta$ -value, the average of free collision or safe navigation is 5 out of 20 trials. This means that the robot experiences more failures by crashing into dynamic obstacles. The percentage of safe navigation is only 25%. Figure 8(c) presents our robot performances to avoid two dynamic obstacles by using SFM with adaptive  $k$ -value. The robot can reach the target by achieving 47 iterations in average for two schemes (#5 and #6). By using SFM with adaptive  $k$ -value, the average of free collision or safe navigation is 18 out of 20 trials. The percentage of safe navigation is the highest with an achievement about 90%. In these experiments, the results were worse than the experiment using static obstacle, because we deliberately direct the dynamic obstacle to block our robot movement by cutting the path right in front of it. Some collisions with obstacle were also caused by the characteristic of the robot dynamic which is very inert when used at high speed.

From Table 6 and Figure 9(b), we show the robot tried to avoid two dynamic obstacles who are moving together towards the same target position by performing obstacle avoidance abilities for different fixed  $k$ -values. The robot can reach the target by achieving 36 iterations in average for two schemes (#1 and #2). By using fixed  $k$ -value, the average of free collision or safe navigation is 19 out of 20 trials. This means that the robot successfully avoids the collision about 95%. Meanwhile, Figure 8(a) shows the robot performances to avoid two dynamic obstacles by using APF with fixed  $\zeta$ -value. The robot can reach the target by achieving 45 iterations in average for two schemes (#3 and #4). By using APF with fixed  $\zeta$ -value, the average of free collision or safe navigation is 9 out of 20 trials. This means that the robot experiences more failures by grazing into dynamic obstacles. The percentage of safe navigation is about 45%. Figure 8(c) presents our robot performances to avoid two dynamic obstacles by using SFM with adaptive  $k$ -value. The robot can reach the target by achieving 38 iterations in average for two schemes (#5 and #6). By using SFM with adaptive  $k$ -value, the average of free collision or safe navigation is 19 out of 20 trials. The percentage of safe navigation is the highest with an achievement about 95%.

In these experiments, the performance of the SFM-based navigation system is still much better than the APF when used to move at high speeds. The responsiveness and security shown by the SFM method are also much better than APF. However, based on both dynamic tests, SFM with adaptive  $k$ -value provides superior performance than SFM with fixed  $k$ -value and APF with fixed  $\zeta$ -value.

Table 6. The experimental schemes for dynamic obstacles where all robots pursue the same target location

Scheme	Avg. Num. of Iterations	Dynamic Obstacles		
		Avg. All	Num. of Free Collision	Percentage of safe navigation
#1	37	36	9 of 10	19 of 20
#2	35		10 of 10	95 %
#3	43	45	4 of 10	9 of 20
#4	47		5 of 10	45 %
#5	40	38	9 of 10	19 of 20
#6	36		10 of 10	95 %

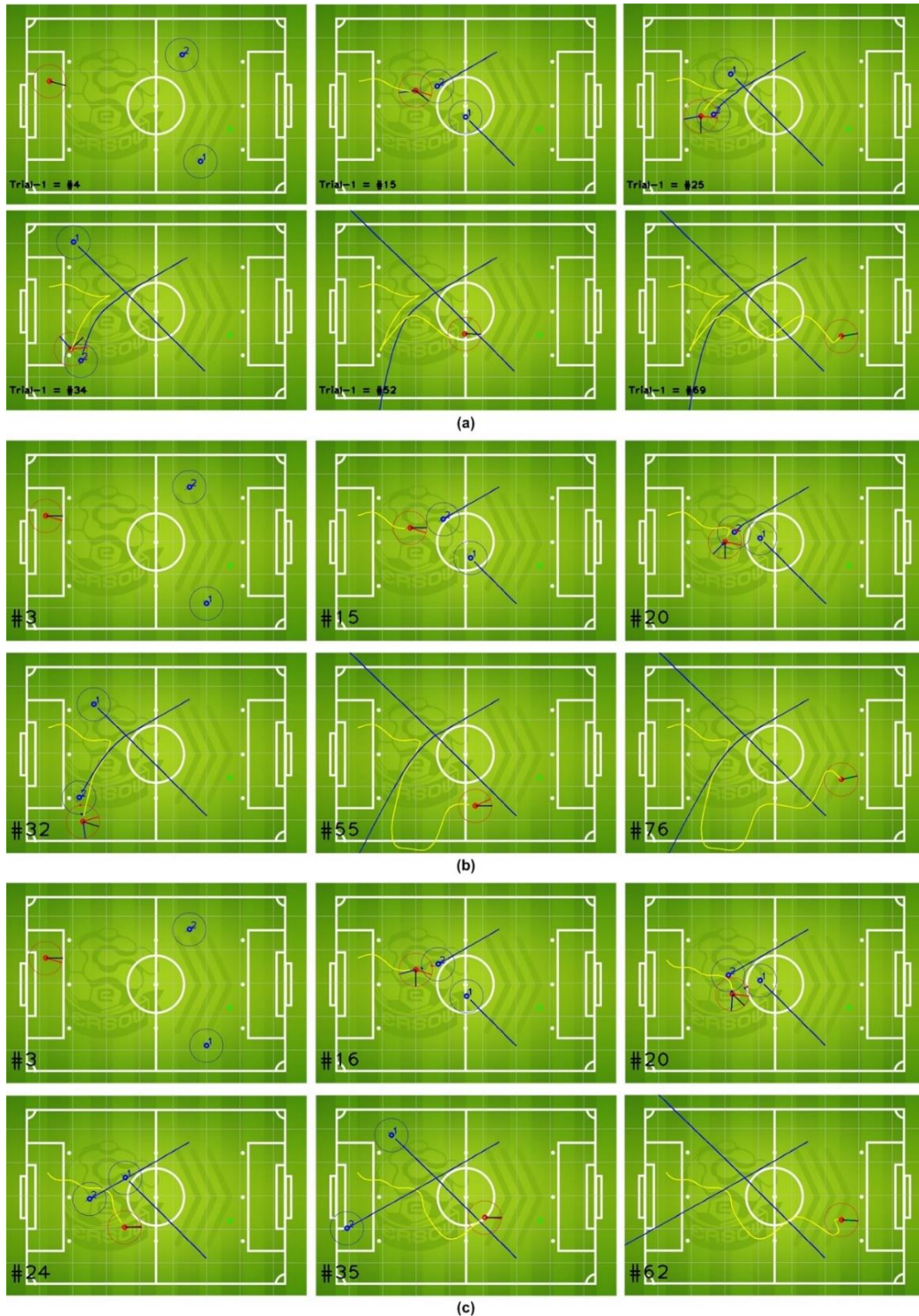


Figure 8. Examples of the robot's responsiveness when avoiding dynamic obstacles who are cutting our robot movement. (a) using APF with fixed  $\zeta$ -value for scheme 3. (b) using SFM with fixed  $k$ -value for scheme 1. (c) using SFM with adaptive  $k$ -value for scheme 5. The blue lines are the trajectories made by the dynamic obstacles or opponent's robots, while the yellow one is the trajectory made by our robot.

### 3.3. Computation time

We conduct computational time testing for each iteration in addition to testing the responsiveness of our robot navigations. Each iteration time was calculated by reading some functions from the procedure, such as defining goal interaction space, robot interaction space, obstacle interaction space, computing forces, and



providing control data to the simulator. When using the fixed  $k$ -value approach, each iteration takes an average of 32.34 msec to complete. The average execution time for FIS to adaptively adjust the  $k$ -value is 35.27 msec. This suggests that employing FIS results in a 9.05% increase in processing time.

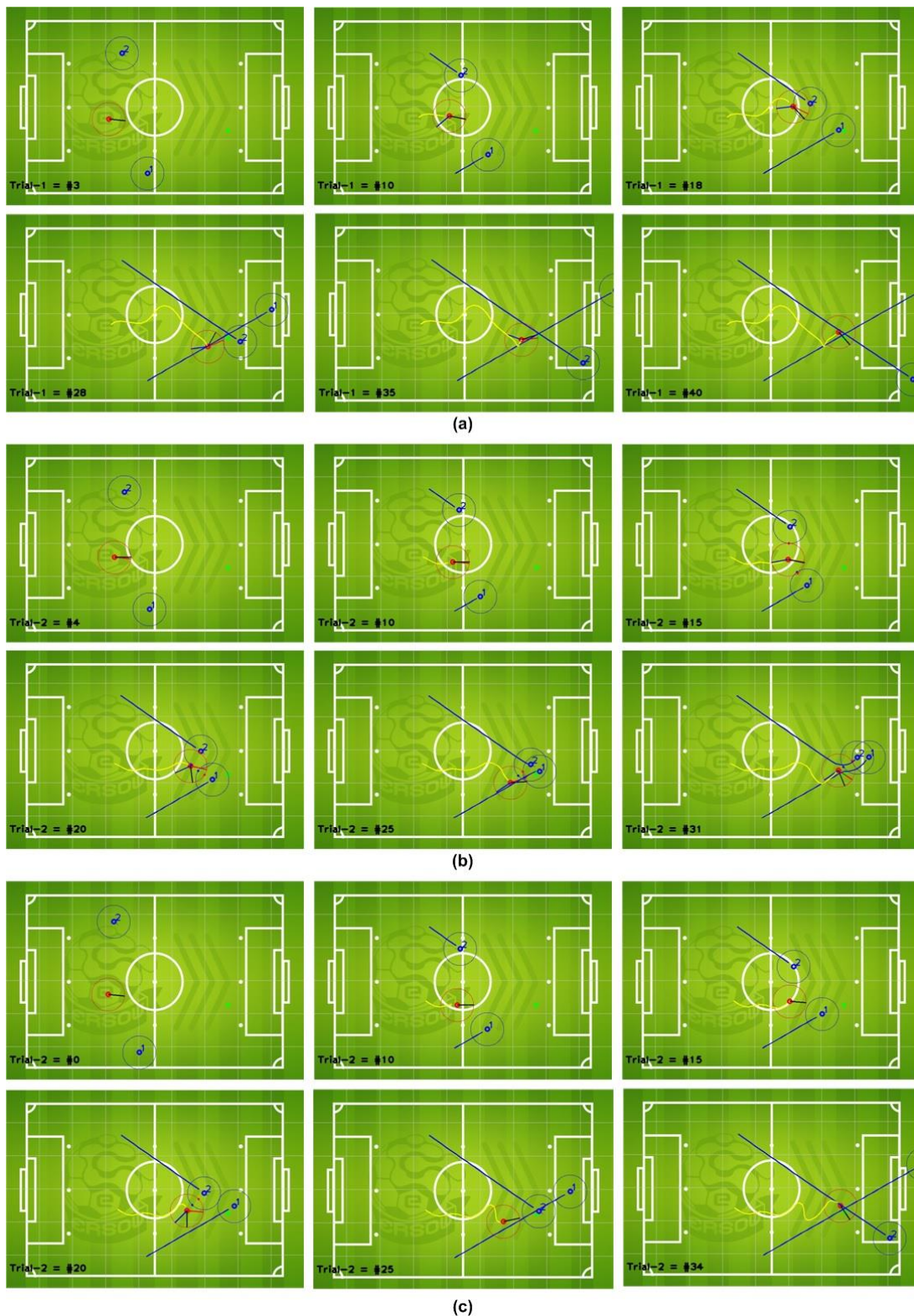


Figure 9. Examples of the robot's responsiveness when avoiding dynamic obstacles who are moving towards the same target position. (a) using APF with fixed  $\zeta$ -value for scheme 3. (b) using SFM with fixed  $k$ -value for scheme 1. (c) using SFM with adaptive  $k$ -value for scheme 5. The blue lines are the trajectories made by the dynamic obstacles or opponent's robots, while the yellow one is the trajectory made by our robot.

#### 4. CONCLUSION

Using a Fuzzy Inference System (FIS) method, we created a novel strategy for adapting the Social Force Model (SFM) parameter. Because the robot must deal with responsiveness while also avoiding obstacles in the field, some modification is required. We assessed and compared our proposed technique to the SFM with fixed  $k$ -value method and APF with  $\zeta$ -value method in this paper to show that our proposed method outperforms the others. Experiments have shown and stressed that adjusting the gain factor,  $k$ , adaptively is a very effective way to control the robot's response. For trials with static and dynamic obstacles, the percentage of safe navigation using the proposed method reaches 100%, 90% and 95%, respectively. The average number of sampling times for static and dynamic obstacles is 60 iterations for static obstacles, while it achieves 47 iterations and 38 iterations for two schemes of dynamic obstacle. The addition of the FIS only adds 9.05% to the computation time. In the future, we will also use Robotic Operating Systems (ROS) or Open Robotic Tool Middleware (OpenRTM) to implement this strategy in our real robot soccer game, allowing the system to run in real time despite the fact that many operations must be handled by the CPU.

#### ACKNOWLEDGMENTS

The first author was financially supported by the Research and Community Service Unit (P3M) of the Politeknik Elektronika Negeri Surabaya.

#### REFERENCES

- [1] B. Damas et.al., "A modified potential fields method for robot navigation applied to dribbling in robotic soccer", *Lecture Notes in Computer Science*, pp. 65-77, 2003.
- [2] P. Zhang et.al., "Soccer robot path planning based on the artificial potential field approach with Simulated Annealing", *Robotica*, vol. 22, pp. 563-566, 2004.
- [3] Q. Zhang Q et.al., "An obstacle avoidance method of soccer robot based on evolutionary artificial potential field", *Energy Procedia* vol. 16, pp. 1792-1798, 2012.
- [4] R. T. U. Albab et.al., "Path planning for mobile robot soccer using Genetic Algorithm", *The 19<sup>th</sup> International Electronics Symposium on Engineering Technology and Applications (IES-ETA)*, pp. 276-280, 2017.
- [5] M. L. Afakh et.al., "Bicycle path planning on omnidirectional mobile robot using Fuzzy Logic Controller", *The 10<sup>th</sup> International Conference on Information Technology and Electrical Engineering (ICITEE)*, pp. 237-241, 2018.
- [6] S. Rodriguez et.al., "Fast path planning algorithm for the robocup small size league", *Lecture Notes in Computer Science* 8992, 2015.
- [7] F. Rahman et.al., "Real-time kinodynamic motion planning for omnidirectional mobile robot soccer using rapidly-exploring random tree in dynamic environment with moving obstacles", *Computer Science – Robotics*, 2020.
- [8] R. H. Abiyev et.al., "Robot soccer control using behaviour trees and fuzzy logic", *The 12<sup>th</sup> International Conference on Application of Fuzzy Systems and Soft Computing (ICAFFS)*, pp. 477-484, 2016.
- [9] R. A. Priambudi et.al., "Dynamic obstacle avoidance on middle size league robot soccer ERSOW using subtargets", *The 3<sup>rd</sup> International Conference on Applied Science and Technology (iCAST)*, 2020.
- [10] D. Helbing and P. Molnar, "Social force model for pedestrian dynamics", *Physical Review*, vol. E 51, pp. 4282-4286, 1995.
- [11] B. S. B. Dewantara at.al., "Fuzzy social force model for robot soccer navigation: a preliminary report", *The 12<sup>th</sup> International Conference on Information Technology and Electrical Engineering (ICITEE)*, 2020.
- [12] I. K. Wibowo et.al., "Ball detection using local binary pattern in middle size robot soccer (ERSOW)", *IEEE International Conference of Computer and Informatics Engineering (IC2IE)*, pp. 29-32, 2019.
- [13] B. S. B. Dewantara and J. Miura, "Generation of a socially aware behavior of a guide robot using reinforcement learning", *IEEE International Electronic Symposium (IES)*, pp. 105-110, 2016.
- [14] C. Yang et.al., "Socially-aware navigation of omnidirectional mobile robot with extended social force model in multi-human environment", *IEEE International Conference on Systems, Man and Cybernetics*, pp. 1963-1968, 2019.
- [15] G. Ferrer et.al., "Robot companion: a social-force based approach with human awareness-navigation in crowded environments", *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1688-1694, 2013.
- [16] F. Zanlungo et.al., "Social force model with explicit collision prediction", *EPL (Europhysics Letters)*, vol. 93, pp. 68005-p1-68005-p6, 2011.
- [17] M. Luber et.al., "People tracking with human motion predictions from social forces", *IEEE International Conference on Robotics and Automation*, pp. 464-469, 2010.
- [18] J. Holland, "Genetics algorithm", *Scientific American*, vol. 267, pp. 66-73, 1992.
- [19] B. S. B. Dewantara, "Building a socially acceptable navigation and behavior of a mobile robot using Q-learning", *IEEE International Conference on Knowledge Creation and Intelligence Computing (KCIC)*, pp. 88-93, 2016.
- [20] R. S. Sutton and A. G. Barto, "Reinforcement Learning: An Introduction", *The MIT Press*, 2012.

- [21] A. Sent et.al., "Simulation of crowd behavior using fuzzy social force model", Proceedings of the 2015 Winter Simulation Conference, pp. 3901-3912, 2015.
- [22] E. Mamdani, "Application of fuzzy logic to approximate reasoning using linguistic synthesis", IEEE Transactions on Computer C, vol. 26, pp. 1182-1191, 1977.
- [23] M. Bliesener et.al., "Robotino", Festo Workbook, 2011.
- [24] E. Hall et.al., "Proxemics", Current Anthropology, vol. 9, pp. 83-108, 1974.
- [25] <http://www.coppeliarobotics.com>, "Vrep: Virtual robot experimentation platform", 2019.

## BIOGRAPHY OF AUTHORS



Bima Sena Bayu Dewantara received the B.Eng. degree in Information Technology from Politeknik Elektronika Negeri Surabaya, Indonesia, and the M.Eng. degree in Electrical Engineering from Sepuluh Nopember Institute of Technology, Indonesia, in 2004 and 2010, respectively. He obtained the Dr.Eng. degree from Toyohashi University of Technology, Japan, in 2016. In 2005, he joined the Department of Electronic Engineering at Politeknik Elektronika Negeri Surabaya, Indonesia, as a lecturer. He moved to the Department of Informatics and Computer Science in 2007. Currently, he is an Assistant Professor in the major field of autonomous intelligent systems which covers pattern recognition, computer vision, machine learning, robotics system, and human-machine interactions. He received The Best Achievement Award from Toyohashi University of Technology, Japan, in 2016 and also received some Best Paper Awards in 2016, 2018, 2019 and 2020.



Bagus Nugraha Deby Ariyadi received his B.Eng. degree in Electronic Engineering from Politeknik Elektronika Negeri Surabaya, Indonesia in 2018. Currently, he is pursuing an applied master's degree at Politeknik Elektronika Negeri Surabaya, Indonesia. His research area is robotics.



Hary Oktavianto received his B.E., M.E., and Ph.D. degrees from the Department of Electrical Engineering of Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia, in 2001, from the Department of Electrical Engineering of National Taiwan University of Science and Technology, Taipei, Taiwan, in 2010, and from the Department of Electrical and Electronic Information Engineering of Toyohashi Technology, Toyohashi, Japan, in 2017, respectively. Since 2002, he has been with the Department of Electrical Engineering of Politeknik Elektronika Negeri Surabaya, Indonesia, as a lecturer. His research interests include embedded systems, computer vision, smart sensor, and optoelectronic integrated circuits.