Neural Fields for Controlling Formation of Multiple Robots

Mohamed Oubbati and Günther Palm

Abstract—In this paper we investigate how neural fields can produce an elegant solution for the problem of moving multiple robots in formation. The objective is to acquire a target, avoid obstacles and keep a geometric configuration at the same time. Several formations for a team of three robots are considered.

I. INTRODUCTION

Recently, there has been increased research interest in systems composed of multiple mobile robots exhibiting cooperative behavior. Such systems are of interest for several reasons: i) tasks may be too complex (or impossible) for a single robot to accomplish, and ii) using several simple robots can be easier, cheaper, and more flexible than having a single powerful robot for one task. This is inspired from the nature. Living in a group allows animals to combine their sensors to maximize the chance of detecting predators, or finding food.

Our problem in cooperative robotics is to maintain a certain geometric configuration during movement. The reason is that there are many interesting tasks that require multiple robots to coordinate their movements, for example box-pushing [1], and movement of a team of military robotic vehicles as a scout unit [2]. There are two main ways of approaching this problem: model-based control, and behavior-based control. In the first, the objective is to build a model for the team of robots and the desired task, and develop a framework to optimise their performance [3]. This method has a limit success when the robots behave in a dynamic and unknown workspace. By contrast, in behavior-based control no model is required. The control system is in a form of reactive behaviors to the current state of the environment [2].

Recently, the theory of dynamical systems has proven to be an elegant and easy to generate robot behavior [4][5][6][7]. The so-called Dynamic Approach invented by Schöner in 1995 [8] provides a framework to design differential equations for so-called behavior variables, which generates the robot’s behavior. Usually, these variables directly parameterize the elementary behavior to be generated. There are cases, however, for which the behavioral variable needs a more general form. For example, a behavioral variable can have multiple values or even no value at all. In those cases, it is necessary to express it by a continuous function. In the dynamic approach, neural fields can represent such variable. Originally, neural fields were proposed by Amari [9] as models of the neurophysiology of cortical processes. They are equivalent to continuous recurrent neural networks, in which units are laterally coupled through an interaction kernel and receive external inputs. In [10], neural fields have been used to recognize complex motion patterns. In [11], they were applied to intelligent cruise control. In robot motion control, neural fields were used for target-acquisition in presence of static obstacles, and for manipulator control [12]. Recently, we used them to navigate the mobile robot to its goal in an unknown environment without any collisions with static or moving obstacles. Furthermore, their competitive dynamics were used to optimize the target path through intermediate home-bases [13].

In this paper we investigate how neural fields can produce an elegant solution for the problem of moving in formation. The objective is to acquire a target, avoid obstacles and keep a geometric configuration at the same time. Several formations for a team of three robots are considered (Fig. 1):

- **line**: robots move line-abreast,
- **column**: robots move one after the other,
- **triangle**: robots move in a triangle formation.

The strategy is to have a leader (robot 1) guiding followers (robots 2 and 3). We assume that the robots are identical, and don’t fail during travel.

In the following, we will first describe the basic concept of neural fields. Then we will present our navigation model and some simulation results.

II. NEURAL FIELDS

Originally, these fields were proposed by Amari [9] as models of the neurophysiology of cortical processes. They are equivalent to continuous recurrent neural networks, in which units are laterally coupled through an interaction kernel and receive external inputs. The field equation of a one-dimensional neural field is given by

\[ \dot{u}(x,t) = \int_{\Omega} J(x,y) u(y,t) \, dy + I(x,t) \]

where \( \dot{u}(x,t) \) is the time derivative of the field variable, \( J(x,y) \) is the interaction kernel, \( I(x,t) \) is the external input, and \( \Omega \) is the domain of the field.
\[ \tau \dot{u}(\varphi, t) = -u(\varphi, t) + S(\varphi, t) + h + \int_{-\infty}^{+\infty} w(\varphi, \varphi') f(u(\varphi, \varphi')) d\varphi' \]

where \( u(\varphi, t) \) is the field excitation at time \( t \) \( (t \geq 0) \) at the position \( \varphi \in \mathbb{R} \). The temporal derivative of the excitation is defined by

\[ \dot{u}(\varphi, t) = \frac{\partial u(\varphi, t)}{\partial t} \]  

(2)

The constant \( h \) defines the pre-activation of the field, and \( f(u) \) is the local activation function. Usually, \( f \) is chosen as a step-function:

\[ f(u) = \begin{cases} 1, & u \geq 0 \\ 0, & u < 0 \end{cases} \]  

(3)

\( S(\varphi, t) \in \mathbb{R} \) represents the stimulus input of the field which is dependent on the field position and varies with time. A nonlinear interaction between the excitation \( u(\varphi) \) at the position \( \varphi \) and its neighboring positions is achieved by the convolution of an interaction kernel \( w(\varphi, \varphi') \). The function \( w(\cdot) \) is, usually, chosen as a Mexican hat function. With this shape, excitatory connections dominate for proximate units, and inhibitory connections dominate at greater distances.

The activation dynamics (1) can have different types of solutions [9]:

1. \( \emptyset \)-solution, if \( u(\varphi) \leq 0, \forall \varphi \).
2. \( \infty \)-solution, if \( u(\varphi) > 0, \forall \varphi \).
3. \( \alpha \)-solution, if there localized excitation from a place \( a_1 \) to a place \( a_2 \). This solution is also called a single-peak or mono-modal solution.

The correct choice of the parameters of the field equation enables the existence of an \( \alpha \)-solution. In this solution, when an input of a stimulus \( S(\varphi, t) \) is very large compared with the within-field cooperative interaction, a single-peak will be stabilized by interaction, even if the stimulus is removed. In navigation control, a single-peak is used as a behavior variable, for example the heading angle.

III. CONTROL APPROACH

A. Field Activation

Each robot 1, 2, and 3 has its own neural field model. We choose the robots’ headings \( \varphi_k \) relative to a world-fixed reference direction as behavioral variables. Hence, each neural field has to encode angles from \(-\pi\) to \(+\pi\). By means of a codebook we use \( N \) discrete directions.

After discretization of (1), each neural field will be:

\[ \tau \dot{u}_k(i, t) = -u_k(i, t) + \sum_{j=1}^{N} w(i, j) f[u_k(j, t)] + S_k(i, t) + h \]

where \( k \in \{1, 2, 3\} \).

The interaction kernel is chosen as:

\[ w(i, j) = k_w e^{-\sigma_w(i-j)^2} - H \]

(5)

where the parameter \( \sigma_w \) fixes the range of excitation, and \( k_w \) its amplitude. The global inhibition \( H \) allows only one localized peak on the field.

After the stabilization of the field, the most activated neuron decodes the direction to be executed by the robot:

\[ \varphi_F = \arg \max_i u(i) \in [1, N] \]  

(6)

In order to update the field stimulus, we assume that each robot has a 360° sensor, with a radius range equal to 1. On each neural field we chose \( N = 60 \) neurons, which means that each direction \( N \) decodes a step of 6°. Before selecting an appropriate direction of a robot, each neural field needs some necessary information (stimulus).

B. Field Stimulus

Based on the tasks described earlier, the stimulus is determined according to two stimulus-functions. These functions describe:

- the direction towards the target. This stimulus is designed excitatory, showing an attraction to the desired direction of each robot.
- the directions to obstacles. In this case each follower has the target stimulus :

\[ S_{Tk}(i, t) = C_{Tk1} - C_{Tk2}|i - i_{Tk}(t)| \]

(7)

where \( C_{Tk1}, C_{Tk2} \) are constants positive, and \( i_{Tk}(t) \) is the field position, equivalent to the main target direction at time \( t \).

The followers are attracted by the required configuration relative to the leader. In this case each follower has the target stimulus :

\[ S_{Tl}(i, t) = C_{Tl1} - C_{Tl2}|i - i_{Tl}(t)| \]

(8)

where \( C_{Tl1}, C_{Tl2} \) are constants positive, and \( i_{Tl}(t) \) is the field position, equivalent to the direction of the required position of the robot \( k \) \( (k \in \{2, 3\}) \) at time \( t \).

- the directions to obstacles \( S_{O}(i, t) \) \( : l \in [1, N_{Obst}] \) where \( N_{Obst} \) is the number of obstacles detected by the robot sensors. This stimulus must be inhibitory, since obstacles collision must be avoided. It is chosen as a Mexican Hat function centered at the direction of an obstacle.

\[ S_{O}(i, t) = C_O e^{-\sigma_O(i-i_l)^2} \]

(9)

where \( C_O \) and \( \sigma_O \) are positive constants. \( \sigma_O \) defines the range of inhibition of an obstacle. In practical situations, this parameter is tuned according to the radius of the robot and the obstacles. \( i_l \) reflects the direction of the obstacle \( l \) at time \( t \). Obstacles here include also other robots, since they should avoid collision among them. However, the stimulus consider only obstacles, which their distances \( d_{O} \) to the robot are below a threshold \( d_{Th} \). Thus, the global obstacle stimulus is determined by:

\[ S_{O}(i, t) = \sum_{l=1}^{N_{Obst}} g(d_O)S_{O}(i, t) \]

(10)
where \( g \) is a step function:

\[
g(d) = \begin{cases} 
1, & d < d_{Th} \\
0, & d \geq d_{Th}
\end{cases} \tag{11}
\]

Using the step function \( g \), only obstacles, whose distances \( d_{oi} \) to the robot are below a threshold \( d_{Th} \), are considered.

The contributions of different stimuli define the global stimulus for each neural field. For the leader neural field, the stimulus at time \( t \) is determined by

\[
S_{\text{leader}}(i, t) = S_{T\text{leader}}(i, t) - C_M S_{\text{O}}(i, t) \tag{12}
\]

For the followers, the stimulus is:

\[
S_k(i, t) = S_{Tk}(i, t) - C_M S_{\text{O}}(i, t) \tag{13}
\]

where \( k \in \{2, 3\} \), and \( C_M \) is a constant positive.

C. Formation Control

For each configuration, two tasks, formation-speed and formation-steer run simultaneously to maintain each robot in its desired position.

1) formation-speed: In a free obstacle situation, the leader moves with its maximum speed \( V_{\text{max}} \), and slows down when it approaches a target. This dynamics can be chosen as:

\[
V_T(t) = V_{\text{max}}(1 - e^{-\sigma_O d_T}) \tag{14}
\]

where \( \sigma_O \) is a constant tuned in relation with the acceleration capability of the robot, and \( d_T \) represents the distance between the robot and the target at time \( t \). When it approaches obstacles, its velocity must be reduced. This dynamics can be chosen as:

\[
V_O(t) = C_O(1 - g(d_{no})e^{-\sigma_{O_o}(\varphi_F - i_{no})^2}) \tag{15}
\]

where \( C_O \) and \( \sigma_{O_o} \) are constant. \( i_{no} \) is the nearest obstacle direction to the robot movement direction, and \( d_{no} \) is its distance relative to the robot. The final dynamics of the leader velocity is the contribution of (14) and (15). It is also considered when no appropriate direction can be selected, for example when the robot is completely surrounded by obstacles. In this case the robot must stop until the environmental situation changes. Thus, the global leader velocity that satisfies the above design criteria is the following:

\[
V_{\text{leader}}(t) = \begin{cases} 
V_T(t)V_O(t), & \varphi_F > 0 \\
0, & \varphi_F \leq 0
\end{cases} \tag{16}
\]

The following steps summarize how a follower selects its speed:

- If a follower is behind its desired position, it should speed up.
- If a follower is in front of its desired position, it should slow down.

In presence of obstacles, the speed is slowed down, depending on the distance to the obstacle. This speed dynamics can be chosen as follows:

\[
V_k(t) = \begin{cases} 
(V_{\text{leader}}(t) + p|\delta_{\text{speed}}|)V_O(t), & \varphi_F > 0 \\
0, & \varphi_F \leq 0
\end{cases} \tag{17}
\]

where \( p \) is a positive constant parameter used to adjust the rate of correction, and \( \delta_{\text{speed}} \in [-1, +1] \) is the correction term. Its value varies depending on how far a follower from its desired position is.

2) formation-steer: According to Fig. 2, desired positions for the configurations line and triangle are:

\[
\begin{align*}
\{ x_2 &= x_{\text{leader}} + \text{dist}_2 \sin(\theta_{\text{leader}} - \alpha_2) \\
y_2 &= y_{\text{leader}} - \text{dist}_2 \cos(\theta_{\text{leader}} - \alpha_2)
\end{align*} \tag{18}
\]

\[
\begin{align*}
\{ x_3 &= x_{\text{leader}} - \text{dist}_3 \sin(\theta_{\text{leader}} + \alpha_3) \\
y_3 &= y_{\text{leader}} + \text{dist}_3 \cos(\theta_{\text{leader}} + \alpha_3)
\end{align*} \tag{19}
\]

where \( \text{dist}_2 \) and \( \text{dist}_3 \) are the distances of the followers 2 and 3 relative to the leader, respectively. In the case of line formation, \( \alpha_2 = 0 \) and \( \alpha_3 = 0 \).

For the column formation, desired positions are:

\[
\begin{align*}
\{ x_2 &= x_{\text{leader}} - \text{dist}_2 \cos(\theta_{\text{leader}}) \\
y_2 &= y_{\text{leader}} - \text{dist}_2 \sin(\theta_{\text{leader}})
\end{align*} \tag{20}
\]

\[
\begin{align*}
\{ x_3 &= x_{\text{leader}} - \text{dist}_3 \cos(\theta_{\text{leader}}) \\
y_3 &= y_{\text{leader}} - \text{dist}_3 \sin(\theta_{\text{leader}})
\end{align*} \tag{21}
\]

IV. Results

The simulations were generated by a software simulator written in MATLAB. Fig. 3 shows a simulation run for the line-abreast formation. At the beginning the followers started with a higher speed, in order to reach their desired positions (Fig. 3 b). When a robot was close to an obstacle, the stimulus contained obstacle entries, which brought the robot away from the obstacle. This behavior was expected, since obstacle-avoidance behavior has the highest priority. The same behaviors can be observed on Fig. 4 for the formation triangle. It shows the case of tracking a moving target with obstacle avoidance. Fig. 5 shows the column formation control in a Door Passing test. In this test, the control approach has successfully provided the robots the ability to traverse through a narrow gap, and maintaining the column formation.
V. DISCUSSION AND CONCLUSION

In this paper, we have demonstrated how neural fields have been chosen as a framework for the behavior-based control for multiple robots, which have no prior information about the environment. While their suitability has been already proven to solve the problem of target-acquisition with obstacle avoidance for one robot [11][12], we described here how neural fields could be used in a more complex problem: tracking static and moving targets, avoiding obstacles, and maintaining formation for multiple robots. The navigation model is developed in order to produce peak-solutions of three fields activations, which encode the appropriate direction of three robots in response to a change in the environment. We have decomposed the problem of formation control into: 1) control of a single lead robot and 2) control of two follower robots in the team. The lead robot has to acquire the main targets, while the follower robots maintain formation. In addition, all robots have to avoid obstacles and collision among them. Based on these desired behaviors, the three neural fields were provided by the necessary stimulus information. Three geometric configurations were considered: line, column, and triangle. For each configuration, two tasks, formation-speed and formation-steer run simultaneously to keep each robot in its desired position. Simulations results demonstrate the smooth behavior of the robots, despite all
imposed constraints. Fig. 4 demonstrates that tracking a moving target, avoiding obstacles, and maintaining triangle formation is not an absolute limitation of our approach. The scenario of passing a door in a column formation did not present any difficulty as well (Fig. 5).

All in all, the neural field approach, described in this paper, provide an elegant solution for behavior-based control of multiple mobile robots. However, a real implementation is still needed to ensure the effectiveness of this approach.

VI. ACKNOWLEDGMENTS

This work was partially supported by the German Academic Exchange Service (DAAD).

REFERENCES