Modeling Complex Emergent Discrete Event Systems: A Case Study In Robotic Swarm Motion

WAQAR MAHMOOD, SAROSH HASHMI
NUST Institute of Information Technology,
National University of Sciences and Technology,
House No. 166-A, Street No. 9, Chaklala Scheme 3, Rawalpindi,
PAKISTAN.

Abstract: - Complex Emergent Discrete Event Systems (CEDES) are those where sequence of local events between simple agents or agents and their environment lead to the emergence of overall behavior following certain design goals. Such systems are proved to be robust and scaleable and therefore find many applications in real world. However they require sound mathematical models that would guarantee desired emergent behavior. We in this paper consider such a system which consists of a swarm of wirelessly connected simple robots and present a novel matrix formulation to model their emergent coherent movement. This approach can be applied not only to model the dynamics of CEDES but also to quickly verify its emergent behavior and is readily executable. The simulation in MATLAB gives interesting insight into emergent behavior of the CEDES over time, which otherwise require expensive experimentation.

Key-Words: - Complex Emergent Discrete Event Systems, Emergent behavior, Mathematical modeling, Swarm robotics

1 Introduction

Efficient modeling of discrete event systems is most important for their control, reconfiguration and optimization. We in this paper consider a class of discrete event systems where agents comprising the system are simple but the sequence of local events between these agents or agents and their environment lead to the emergence of desired complex behaviors and hence we call such systems as ‘Complex Emergent Discrete Event Systems’. Comprehensive modeling methodologies need to be developed to engineer such systems with high degree of dependability and reliability thereby guaranteeing desired collective emergent behavior, preventing undesirable behavior and determining parameters that optimize overall system performance. Robotic systems based on swarm intelligence, drawing inspiration from biological domain examples of social insects (like ants, birds etc.), are examples of such systems [1]. The individual actions of these autonomous entities lead to collective emergent behavior, that needs to be validated against set goals thereby necessitating a sound integrated modeling approach for specifying, proving and testing individual robots as well as overall DES emergent behavior. One approach could be the application of matrices to specify and verify the behavior of such systems. Matrix formulation has been shown to be successfully employed for modeling, simulation and analysis of complex DES like modern flexible manufacturing systems with reentrant flow lines [2]. It enables fast design and reconfiguration of rule based controllers for such manufacturing systems and state equation of these controllers give comprehensive dynamical description of these DES. In this paper we will use a matrix formulation to model the behavior of a complex emergent discrete event system consisting of a swarm of wirelessly connected simple robots.

This paper is organized as follows: Section 2 gives a review on the state of art in complex emergent discrete event systems modeling, Section 3 defines and applies our proposed matrix formulation to the case study of a wireless connected robotic swarm, Section 4 states details of emergent behavior observed on simulating the proposed model in MATLAB and finally section 5 gives future directions and concluding remarks.

2 Literature Review

A few attempts have been made by different researchers to model complex emergent behavior of discrete event systems. A non-spatial probabilistic
model of distributed manipulation tasks by a swarm of robots was presented in [3],[4] based on probabilities calculated by geometrical analysis of the objects involved like, walls, other robots, sticks to be pulled out etc. While their methodology has the advantage of not being relying on computation of orientation and sensory information of individual robots; it is not applicable in scenarios where the arena is overcrowded with robots or for robots with individual sophisticated capabilities like wireless communication. These limitations prevent the use of the methodology to our case study which uses local wireless communication for swarm coherence.

Lerman et al [5],[6] worked on characterizing each agent behavior as being stochastic and Markovian and proposed a stochastic master equation from which rate equations were derived describing the change at macroscopic level of swarm over time. The approach was applied to a scenario of coalition formation of agents in an electronic marketplace where rates of agents joining or leaving coalitions were modeled as function of utility gain of becoming the member of a particular coalition, implying that agents will probably join larger coalitions with better utility gain rather than smaller ones. The methodology was also applied to study the behavior of a group of robots involved in a foraging task to collect pucks from an arena and deliver them to a specified location, with rate equations giving number of robots in searching, homing and avoiding states at a particular instant. However again this methodology is non-spatial and is not applicable to the cases where specification and verification of design goals is dependent on robots locations.

Rouff et al [7],[8],[9] after evaluating various formal methods proposed a fusion of Communicating Sequential Processes (CSP) [10], Unity logic, Weighted Synchronous Calculus of Communicating Systems (WSCCS) [11] and X-Machines [12] to develop an integrated formal method for specifying and verifying emergent behaviors of swarms of large number of cooperating spacecrafts for NASA’s future space exploration missions.

Recently Winfield et al [13] have considered temporal logic for specifying robotic swarm behavior and possibly using a temporal monodic prover [14] to verify probable emergent behaviors, which could be very resource intensive to implement. Stability analysis of a swarm of autonomous cooperative unmanned air vehicles is carried out by calculating eigenvalues of system matrix in [15], however simulation is being employed for analysis when obstacles like walls, other vehicles and communication between vehicles is taken into account.

3 Matrix Formulation

We model the emergent behavior of a wirelessly connected swarm presented by Winfield et al in [13] which uses the simplest ‘alpha algorithm’ of all algorithms presented in [16] for swarm coherence. The finite state machine of individual robot in the swarm is shown in figure below which illustrates that a robot has three states in actual but we neglect the ‘avoidance’ state here for simplicity and thus model the robots to have following two states:

1. Forward State (Abbreviated as Fw)
2. Coherence State (Abbreviated as Coh)

![Fig-1 Finite State Machine of Individual Robot in Swarm](Image)

The alpha algorithm states that by default at each time step, the robots move ‘d’ units in the forward direction on a grid space and their orientation can have values from the set of North, South, East and West only. A robot can also turn left 90°, turn right 90° or take 180° u turn. If direction of robot at next step will be north, its x coordinate is kept constant while y coordinate is increased d units, if next direction is south, again x coordinate is kept same while y coordinate is decremented d units, if next direction is east then x coordinate is increased d units while y coordinate is kept constant and finally if next direction is west, the x coordinate is decreased d units keeping y coordinate constant.

A variable ‘α’ specifies the minimum number of neighbors (connections) which a robot has to maintain, falling below which the robot, is forced to take a 180° turn and change to coherence state. For example if α = 1, it means that loss of last connection with another robot in the swarm triggers the coherence state. If the number of neighbors increase above α the robot takes right or left turn to avoid swarm collapsing on itself.
A robot is said to be connected to another robot, if later is in its connectivity radius ‘r’ and likewise a robot is said to be connected (abbreviated as ‘con’) to the swarm if it is connected with at least $\alpha$ other robots and said to be disconnected otherwise (abbreviated as ~con).

The state transition functions are:
1. $Fw + con \rightarrow Fw$
2. $Fw + ~con \rightarrow$ turn $180^\circ \rightarrow Coh$
3. $Coh + ~con \rightarrow Fw$
4. $Coh + con \rightarrow$ Left or Right turn $\rightarrow Fw$

Let the x, y coordinates of the four robots be $(x_1, y_1), (x_2, y_2), (x_3, y_3)$ and $(x_4, y_4)$ respectively.

We generate a proximity matrix $A$ by finding the Euclidean distances between the respective robots:

$$
A = \begin{bmatrix}
R1 & R2 & R3 & R4 \\
R1 & 0 & a_{12} & a_{13} & a_{14} \\
R2 & a_{21} & 0 & a_{23} & a_{24} \\
R3 & a_{31} & a_{32} & 0 & a_{34} \\
R4 & a_{41} & a_{42} & a_{43} & 0
\end{bmatrix}
$$

Where $a_{ij} = $ Euclidean distance between robot $i$ and robot $j$.

As stated earlier, let $r = $ maximum connected distance. We subtract the proximity matrix $A$ by a matrix $B$ having all elements equal to $r$ except the diagonal ones, resulting in matrix $C$:

$$
C = \begin{bmatrix}
R1 & R2 & R3 & R4 \\
R1 & 0 & b_{12} & b_{13} & b_{14} \\
R2 & b_{21} & 0 & b_{23} & b_{24} \\
R3 & b_{31} & b_{32} & 0 & b_{34} \\
R4 & b_{41} & b_{42} & b_{43} & 0
\end{bmatrix}
$$

Converting the negative values into zeros and positive values into ones of the above resultant matrix $C$ gives a matrix $D$ which has all binary values, where:

$$
d_{ij} = \begin{cases} 
0 & \text{if robot } i \text{ and } j \text{ are connected and } \\
1 & \text{otherwise.}
\end{cases}
$$

D (Binary values matrix) =

$$
\begin{bmatrix}
0 & d_{12} & d_{13} & d_{14} \\
d_{21} & 0 & d_{23} & d_{24} \\
d_{31} & d_{32} & 0 & d_{34} \\
d_{41} & d_{42} & d_{43} & 0
\end{bmatrix}
$$

If we assume a value of $\alpha = 1$ so that the loss of single connection to the swarm triggers the coherence state. In other words if there are less then two zeros in any row of the above matrix then the respective robot is not connected. Therefore, we can reduce the above matrix $D$ into a column matrix ‘Connect’ by writing ‘0’ in the row if $(\alpha + 1)$ values in the corresponding row of matrix $D$ are zeros and ‘1’ otherwise:

$$
Connect^a = \begin{bmatrix}
\text{connect}_{i1} \\
\text{connect}_{i2} \\
\text{connect}_{i3} \\
\text{connect}_{i4}
\end{bmatrix}^n
$$

Where:

$$
\text{connect}_{i1} = \begin{cases} 
0 & \text{if robot } i \text{ is connected to the swarm, and} \\
1 & \text{otherwise}
\end{cases}
$$

$n$ is the time variable indicating discrete time instance.

We define a column matrix ‘State’ where:

$$
State^a = \begin{bmatrix}
\text{state}_{i1} \\
\text{state}_{i2} \\
\text{state}_{i3} \\
\text{state}_{i4}
\end{bmatrix}
$$

We compute next state vector $State^{n+1}$ by the following equation:

$$
\text{State}^{n+1} = (\text{State}^a \land Connect^a) \oplus Connect^a 
$$

Note: $\oplus$ denotes the ‘XOR’ operation

Following table describe the transition from $nth$ state to $(n+1)th$ state using the above equation:

<table>
<thead>
<tr>
<th>State</th>
<th>Connect</th>
<th>State</th>
<th>State $^{n+1}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
Finally, we define a Direction matrix as follows:

\[
\text{Direction} = \begin{bmatrix}
\text{direction}_{11} & \text{direction}_{12} \\
\text{direction}_{21} & \text{direction}_{22} \\
\text{direction}_{31} & \text{direction}_{32} \\
\text{direction}_{41} & \text{direction}_{42}
\end{bmatrix}
\]

Where:

- direction_{ij} = 0 & direction_{ii} = 0 if robot_i is pointing ‘North’
- direction_{ij} = 1 & direction_{ii} = 1 if robot_i is pointing ‘South’
- direction_{ij} = 0 & direction_{ii} = 0 if robot_i is pointing ‘East’
- direction_{ij} = 1 & direction_{ii} = 1 if robot_i is pointing ‘West’

Further, we combine State^n and Connect^n matrices to form State-Connect augmented matrix and convert any row of this augmented matrix containing all ones to all zeros. Lastly we ‘XOR’ this modified State-Connect augmented matrix with Direction^n matrix to give Direction^{n+1}.

\[
\text{Direction}^{n+1} = \text{State-Connect} \oplus \text{Direction}^n \tag{2}
\]

It may be noted that the conversion of any row in State-Connect matrix containing all ones to all zeros does not affect the behavior of system as the next action to be taken in both cases is same (i.e. to move in forward direction)

Following table describe the transition from nth direction to (n +1)th direction using the above equation:

<table>
<thead>
<tr>
<th>State Transition Function</th>
<th>State-Connect</th>
<th>Direction^n</th>
<th>Direction^{n+1}</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fw + con (\rightarrow) Fw</td>
<td>0 0 0 0 0 0</td>
<td>0 1 0 1 0 1</td>
<td>1 1 1 1 1 1</td>
</tr>
<tr>
<td>Fw + ~con (\rightarrow) turn 180° (\rightarrow) Coh</td>
<td>0 1 0 0 0 0</td>
<td>0 1 0 0 1 1</td>
<td>1 1 1 0 1 0</td>
</tr>
<tr>
<td>Coh + ~con (\rightarrow) Fw</td>
<td>1 1 State-Connect entries converted to ‘0 0’. See entries against ‘0 0’</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

We explain here the modeling of state transition no. 4. The last four rows of the above table state that if a robot is in coherence state ‘1’ and connected to swarm ‘0’ and if its current direction is north ‘0 0’ it will turn right to change its direction to east ‘1 0’. If the robot is in south direction ‘0 1’, it will turn right to change its direction to west ‘1 1’. If the robot is in east direction ‘1 0’, it will take a left turn to change its direction to north ‘0 0’ and finally if the robot is currently in west direction ‘1 1’ it will turn left to south direction ‘0 1’.

In other words, the random behavior of state transition number 4 described earlier is modelled through computing the next direction of robot based on its current direction.

The model presented above is the macroscopic model of the whole swarm. To derive microscopic model and capture the dynamics of individual robots we have to modify these matrices accordingly, like the proximity matrix for robot 1 can take the shape of the following row matrix:

\[
\text{Proximity R1} = [0 3 1 100]
\]

(i.e. the first row of proximity matrix for the whole swarm)

The robot will fill the matrix values by locally sensing within its connectivity radius and finding the Euclidean distances for whatever other robots it finds in the neighbourhood. For the robots which it doesn’t sense any information, it simply put a predefined infinity value like ‘100’ in the above case in order to facilitate the later computation of Connect matrix. Similarly all other matrices are modified accordingly.

4 Results

We validated our model by simulating it in MATLAB and observed the following emergent behaviors of the system:
The swarm remained coherent and move forward at a fast pace provided the initial states and directions of all robots were same at start.

If randomness is introduced in the initial states and directions of individual robots, the swarm divides itself into sub swarms depending on the value of alpha. In such case there may be sub swarm formation of different sizes.

Increasing the value of alpha may increases the size of sub swarms and decreases their number but a very high alpha value results in a very over-reactive swarm, which just remains occupied in rearranging itself and never moves forward.

5 Conclusion and Future Work

A control theoretic approach based on matrices is presented to model the emergent behavior of a CEDES comprising of a wirelessly connected swarm of robots. We modeled important parameters of the system like connection status of robots, their state and direction as individual matrices and then presented linear equations to model the relationship between these parameters. The simulation of model in MATLAB elucidated emergent behavior of DES over time. This experience makes us believe that the approach has potential to develop as a generic design methodology for modeling emergent discrete event systems.

In future we intend to incorporate further control theoretic concepts like controllability and observability in our model by introducing an ‘observer’ that would help in avoiding swarm robots collision and introducing a ‘control’ input to move swarm in the desired direction and control its motion. Finally we intend to introduce soft (fuzzy) angles for robot turns to make it more flexible and capable of complex maneuvers.

5 Acknowledgment

This work was supported in part by the Higher Education Commission, Pakistan.

References:
Dept of Computer Science, Sheffield University, UK.


