2nd International Conference on Autonomous Robots and Agents
December 13-15, 2004 Palmerston North, New Zealand

# State Transition Based (STB) Role Assignment and Behaviour Programming in Collaborative Robotics

G. Sen Gupta[1,2], C. H. Messom [3], S. Demidenko[4]
[1]IIST, Massey University, Palmerston North, New Zealand
[2]School of EEE, Singapore Polytechnic, Singapore
[3]IIMS, Massey University, Albany Campus, New Zealand
[4]School of Engineering & Science, Monash University, Kuala Lumpur, Malaysia

## Abstract

Complex distributed robot control systems consist of mechanical systems powered by actuators that are under the control of computer systems that rely on sensor input, such as vision, touch and torque sensing. Often these systems are installed in production processes in which they must cooperate and collaborate with both humans and other robotic systems. This complexity requires a framework in which the many interacting components can be managed. This paper extends the hierarchy of the state transition based (STB) techniques for managing the behaviour of collaborative robotic systems.

**Keywords**: Robot Colony, State Transition Based (STB) behaviour control, Role Selection, Collaborative robotics

## 1 Introduction

In collaborative robotics, a group of robots operate collectively and cooperate to achieve a common goal or task. Such systems have found their way into production floors, home automation and entertainment robotics. Distributed systems are inherently more complex than individual isolated systems. The complexity is introduced by the various interactions that can occur between the distributed components. These interactions include explicit communication via message passing as well as implicit communication via physical interaction and resource contention. Managing this complexity is one of the significant challenges of distributed robotic systems. As low-level control problems are solved, managing the high level supervisory control of behaviour becomes the main focus of attention.

Most techniques that have been proposed for managing complexity can be categorised as hierarchical systems in which the detailed complexity is abstracted away in subsystems that are considered to be primitive or atomic. The advantage of this approach is that at the higher level the complexity of the problem is greatly reduced and so suitable for automatic or even manual solution. A disadvantage with rigorous abstraction using hierarchical approaches is that the optimal solution may be hidden by an inappropriate hierarchical decomposition.

In managing complexity, a simplified reactive architecture is often adopted. In this approach, instinctive behaviour is developed as reactions to the current sensor input, requiring little modelling of the environment in terms of any internal state [1]. [2] proposes a layered disclosure technique that allows the software agents in the system to act and explain their actions at several levels of abstraction. An effective way to implement agent behaviour is by using State Transition Based Control (STBC) technique [3]. Advantages of this approach include data and process abstraction, which aids development of new functionality.

In this work we extend this technique to build hierarchy of states for complex behaviour and role selection in multi-agent collaborative robotics. In the last decade robot soccer has been extensively used as a test bed for adaptive control of dynamic systems in a multi-agent collaborative environment [4]. Because of the dynamics and high complexity of the robot soccer system it is an ideal platform to test the methodologies proposed in this paper.

## 2 Hierarchy of States

Complex behaviour cannot easily be reduced to simple primitive actions. Normally the action that is required in each state is itself a complex behaviour. For example the *Clear Ball* action in the goalkeeper behaviour [3] cannot simply be to kick the ball, since this might result in scoring an own goal. Similarly the goalie must be careful in approaching the ball so that the ball is not accidentally knocked into its own goal. This can be achieved by creating a state based *clear ball* action as shown in Figure 1.

The state transitions between S0, S1and S2 must be defined and the actions required in each state, taking care to set appropriate sensor and behavioural hysteresis factors.

The actions "*Kick Ball*" and "*Go to safe position*" are low level behaviours implemented in a similar manner to [5]. Since each robot potentially has a large hierarchy of behavioural states, a single state variable is not enough; rather an array of states and sub-state variables must be maintained so that the robot can keep a record of what the current behaviour should be. A consequence of this hierarchy is that when a state transition occurs at a higher level, the sub-states would have to be initialised, since the sub-state behaviours are not normally carried over higher state transitions.

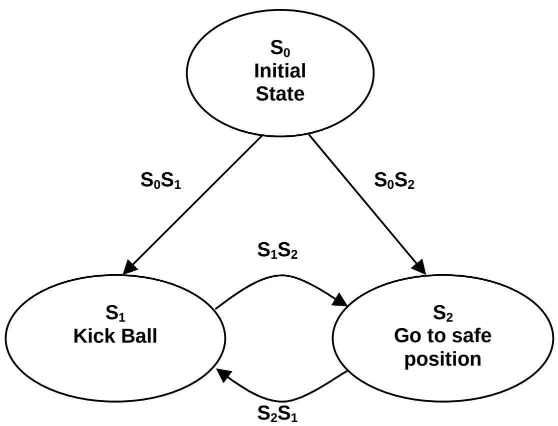


Figure 1: State transition diagram of the *Clear Ball* action.

## 3 Role Selection

This section describes the *Role Selection Layer* for two collaborating robots. In some applications it may be necessary to assign fixed roles to each robot, but in the case of robot soccer where the robots are physically homogenous it is possible to allocate roles (goalkeeping, defence and offence) dynamically.

Figure 2 shows the regions of the field in which the two robots both adopt defensive behaviours (**DD**), others where both adopt attacking behaviours (**AA**) and yet others where one acts defensively and the other attacks (**AD** and **DA**).

The selections of the role that the robots are to adopt are determined by the state transition diagram shown in Figure 3. The roles adopted by the two robots either as an attacker (**A**) or a defender (**D**) is given by the position of the ball. The shaded region around the horizontal centre line defines the hysteresis for transition between **AD** and **DA** behaviours. In this example one robot is statically assigned the goalkeeper role. However, this formulation can be extended to the goalkeeper and it can also be dynamically assigned roles.

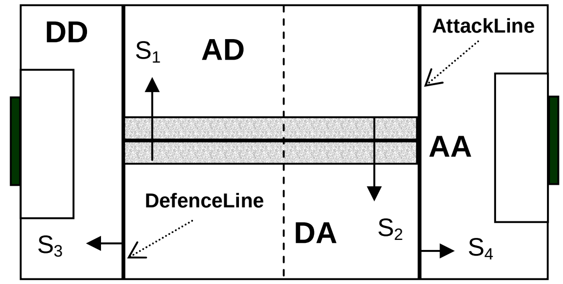


Figure 2: Boundaries for robot *Role Selection*

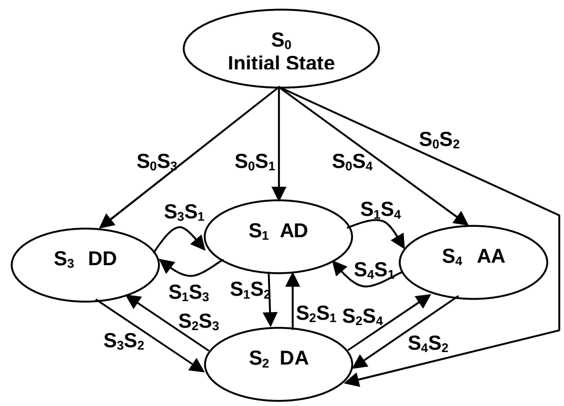


Figure 3. State transition diagram for *Role Selection*

The transitions between states are given by the movement of the ball across the region boundaries specified in Figure 2. However a sensor hysteresis is applied to the boundaries as well as a behavioural hysteresis that allows the attack or defence action to complete.

### 3.1 Defensive Behaviour

Figure 4 shows the regions of the field in which the defensive robot adopts a support goalie (**SG**) behaviour, a defending behaviour (**D**) and a support attacker (**SA**) role.

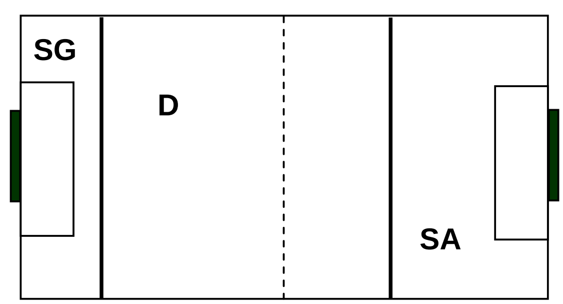


Figure 4. Boundaries for *Defensive Behaviour*

Figure 5 shows the state transition diagram of the defensive behaviour. As well as the **SG**, **D** and **SA** behaviours the state transition diagram shows a backing off (**BO**) behaviour. The transitions between state **SG**, **D** and **SA** are determined by the position of the ball as shown by the boundaries in Figure 4. A sensor hysteresis term is added to these transitions. The additional **BO** state is added so that there is no conflicting condition in which both the robots are trying to kick the ball. The transition $S_4S_2$ occurs when the other robot has the best shoot or pass position on the ball, while the $S_2S_4$ transition occurs when this robot has the best shoot or pass position. In this way any behavioural conflict is resolved.

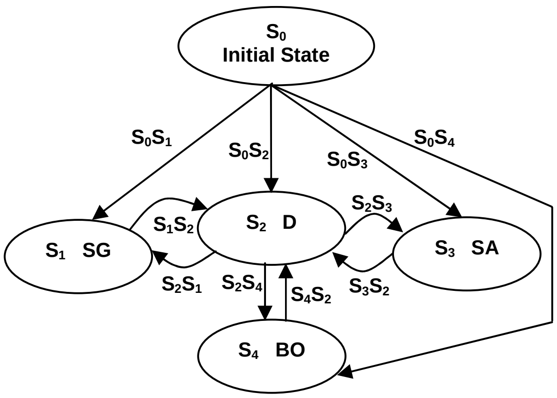


Figure 5. State transition diagram of the *defensive behaviour*

The **SG** state is a defensive state in which the robot idles in a position to support the goalkeeper. Depending on the position of the ball, this may be in line with the ball, waiting for the goalkeeper's pass, or away from the ball allowing the goalkeeper to kick out the ball.

The **SA** state allows the defender to support the action of the attacker, this will normally be a position behind the attacking robot and the ball.

## 3.2 Offensive Behaviour

Figure 6 shows the region of the field where the offensive robot adopts a support goalie and defender (**SGD**) behaviour, shoot behaviour (**Shoot**) and a leave ball (**L**) behaviour.

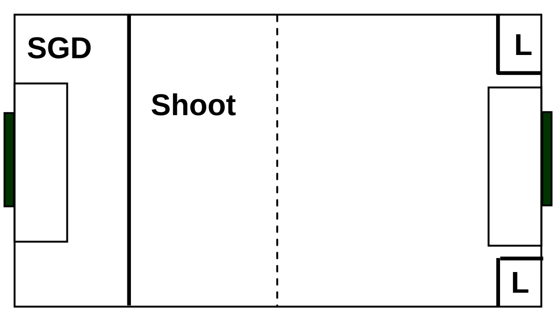


Figure 6. Boundaries for *Offensive Behaviour*

Figure 7 illustrates the offensive behaviour state transition diagram. Similar to the defensive behaviour, the offensive behaviour makes use of a sensor hysteresis term between the **SGD**, **Shoot** and **L** states. A behavioural hysteresis term inhibits the transitions from the backing off (**BO**) state. This ensures that two or more robots do not adopt conflicting actions while shooting or passing the ball.

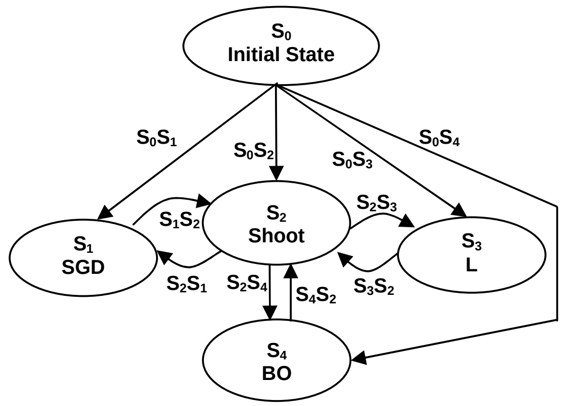


Figure 7. State transition diagram of the *offensive behaviour*

The **SGD** state is similar to the defender's **SG** state except the attacking robot takes a position which is clear of both the goalkeeper and the defender allowing either of them to clear the ball. Its role in some positions is to await a pass from the defender or goalkeeper and in other positions to disrupt the attacking flow of the opposition.

The **L** state is a situation in which the attacking robot is unlikely to score from. In this position the attacking robot idles allowing the opponents goalkeeper or defender to try to clear the ball. This is a favourable position in which to tackle the defenders and steal the ball.

### 3.2.1 The *shoot* action

The *shoot* action of the attacking robot, as shown in Figure 7, is itself a complex task represented by several states and actions. The robot can actually shoot the ball in the opponent's goal only when it is in a good position behind the ball. At times, like when the ball is very close to the walls of the field, it is impossible for the robot to attain a good shooting position. In such a situation, it must dribble the ball along the wall. Another situation is when the robot has crossed the ball and gone past it. It then must attempt to go to a good position behind the ball. So that, in trying to reach a good position, it does not kick the ball towards its own goal, a suitable side position must be reached first. The complete shoot action may thus be represented by a lower level state transition diagram as shown in Figure 8. This illustrates the importance of state hierarchy and the ease with which it helps to implement complex behaviours.

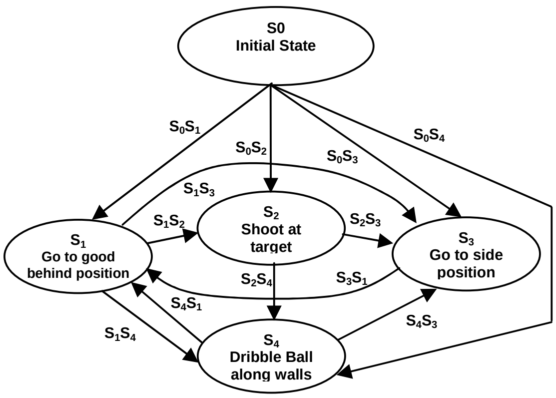


Figure 8. State transition diagram of the *shoot* action

Some of the state transition conditions are as follows-

$S_0S_2$ – Robot behind the ball and in good orientation to shoot towards target

$S_0S_3$ – Robot missed and past the ball

$S_0S_4$ – Calculated *behindPosition* is outside the walls

$S_0S_1$ - $!S_0S_2$ && $!S_0S_3$ && $!S_0S_4$

### 3.2.2 Transition condition for *Backoff* state

For organised collaborative behaviour, and to prevent both the agents from going for the ball at the same time, a decision has to be made about which robot should back off. The simplest case is when both the robots are behind the ball; the robot farthest from the ball will back off and the one nearest to the ball will take the shot at the target, which is the opponent's goal centre.

In Figure 9, both the robots are behind the ball. Since Robot 1 is closer, it will assume the role of a shooter while Robot 2 will go to '*Backoff*' state.

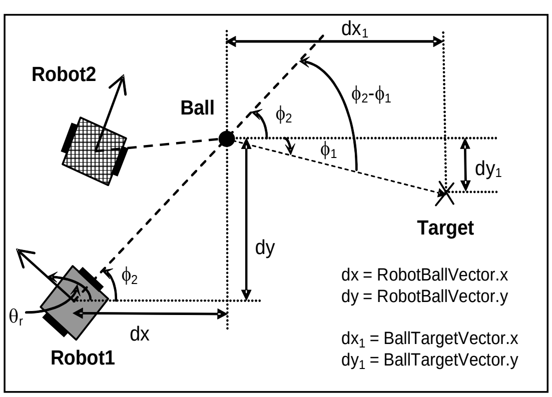


Figure 9. Both the agents behind the ball

The angles shown in Figure 9 are:

$\phi_1$ - *TargetAngle*
(negative value in the case under consideration)
$\phi_2$ - *BallAngle* (positive value)
$\phi_2$-$\phi_1$ - *BallTargetAngle*
$\theta_r$ - *RobotAngle*

The decision is more complex when both or one of the robots is on the wrong side of the ball, i.e. between the ball and the target. Such a situation is shown in Figure 10.

If a robot has BallTargetAngle greater than $90^o$ or less than $-90^o$, it is deemed to be not in a favourable orientation for shooting and hence its distance from the ball is 'penalised' by increasing it by a factor, WRONGSIDEDISTANCEFACTOR. The modified distance of the robot to the ball is then compared with the other robot and the one farthest will back off.

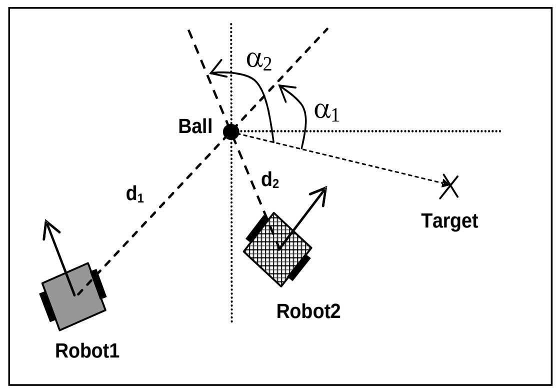


Figure 10. Robot 2 on the wrong side of the ball

In Figure 10, $\mathbf{d_1}$ is the distance of Robot1 to the ball and $\mathbf{d_2}$ is the distance of Robot2 to the ball. Robot 2 has a *BallTargetAngle* ($\alpha$**2**) which is greater than $90^o$. Hence its modified distance to the ball is d1+ WRONGSIDEDISTANCEFACTOR. Since the *BallTargetAngle* of Robot1 ($\alpha$**1**) is less than $90^o$, it is not penalised. So even though physically Robot2 is closer, it will go into '*Backoff*' state and Robot1 will take a shot at the target.

## 4 The DD Role

The DD behaviour state, in which both the robots exhibit defensive behaviour, was specially designed to provide solid defence and make it extremely difficult for the opponent to score. This includes adopting strategic positions around the ball simply to prevent the opponent's robots to physically come near the ball, essentially blocking them out.

The state transition diagram for the DD Role is shown in Figure 11.

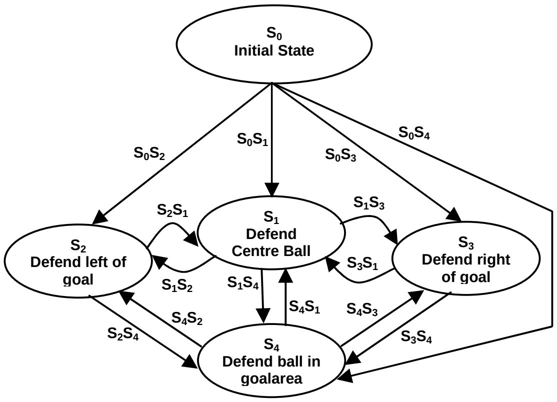


Figure 11. State transition diagram of the *DD Role*

### 4.1 $S_1$: Defending in front of goal area

When the ball is in front of the goal area, as shown by the grid, the two robots take up position on either side of the ball, in close proximity, leaving little space for the opponent's robots to reach the ball without physically pushing the home team robots. Smart calculations ensure that the robots nearest to the left and right positions are positioned accordingly. This situation is depicted in Figure 12.

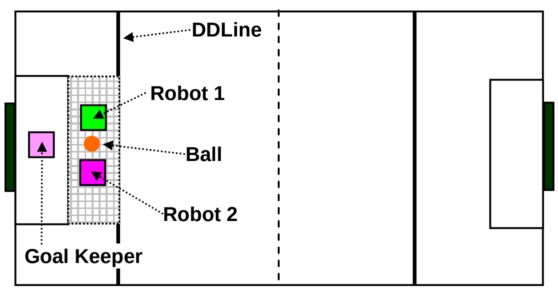


Figure 12. Defending in front of goal area

### 4.2 $S_2$: Defending left of goal area

The positions taken by the home team robots when the ball is to the left of the goal area is shown in Figure 13. The robot nearest to the ball will try to reach it while the other will position itself near the top corner of the goal area. This also prevents the opponent's robot to gain easy access to the ball and push in towards the goal.

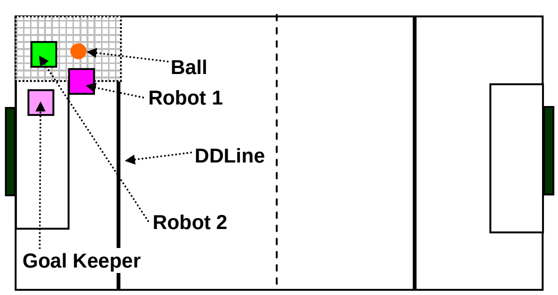


Figure 13. Defending left of goal area

### 4.3 $S_3$: Defending right of goal area

The positions taken by the home team robots when the ball is to the right of the goal area is shown in Figure 14. The robot nearest to the ball will try to reach it while the other will position itself near the bottom corner of the goal area. This also prevents the opponent's robot to gain easy access to the ball and push in towards the goal.

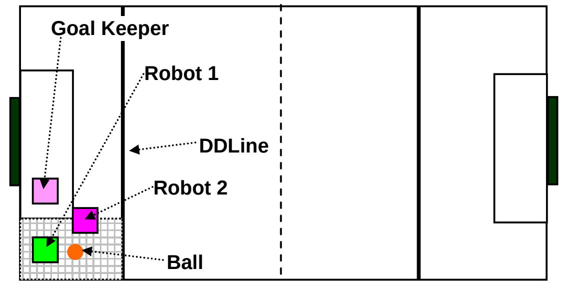


Figure 14. Defending right of goal area

### 4.4 $S_4$: Defending with ball in goal area

When the ball enters the goal area, the goal keeper will take actions to clear it. The positions taken by the other two home team robots are very similar to one when the ball is in front of the goal but outside the goal area. This is shown in Figure 15.

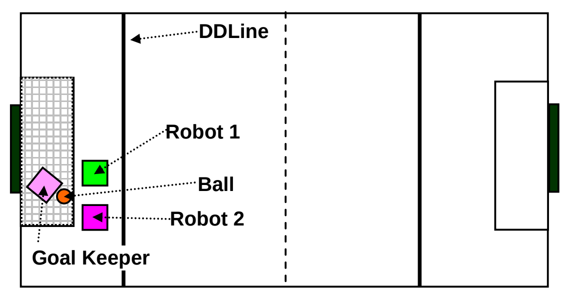


Figure 15. Defending with ball in goal area

## 5 Conclusions

This paper extended the state transition based system (STB) for implementing complex behaviour and strategy for control of distributed robotic systems. The complexity of the system can be greatly simplified and easily managed by building a hierarchy of state transition diagrams. This approach can be used to rapidly prototype new behaviours, which can then be optimised over time.

This STBS technique can also be applied to industrial robotic systems, particularly collaborative systems with highly variable, possibly uncertain environments, such as robots in agricultural automation and processing industries.

## 6 References

[1] Brooks R.A., "A robust Layered Control System for a Mobile Robot", IEEE Journal of Robotics and Automation, vol 1, 1986, pp 1-10

[2] Riley P., Stone P., Veloso M., "Layered Disclosure: Revealing Agents' Internals", Seventh International Workshop on Agent theories, Architectures and Languages, 2000.

[3] G. Sen Gupta, C. H. Messom, S. Demidenko, Lim Yuen Siong, "Identification and Prediction of a Moving Object Using Real-Time Global Vision Sensing", Proceedings of 20th IEEE Instrumentation and Measurement Technology Conference, IMTC 2003, Vail, USA, pp 1402-1406

[4] C. H. Messom, "Robot Soccer – Sensing, Planning, Strategy and Control, a distributed real time intelligent system approach", AROB, Oita, Japan, 1998, pp. 422 – 426.

[5] Thomas P., Springer G., Stonier, R.J. and Wolfs P., "Boundary-avoidance and attack strategy in robot soccer", Fira Robot World Cup France '98 Proceedings.