A Hybrid Control Approach to Autonomous Navigation in Cooperative Multi-Robot Systems using Kripke Models

Suresh Jeyaraman, Antonios Tsourdos, Rafal Zbikowski, and Brian White

Royal Military College of Science, Cranfield University, Shrivenham SN6 8LA, United Kingdom.

{S.Jeyaraman, A.Tsourdos, R.W.Zbikowski, B.A.White}@rmcs.cranfield.ac.uk

Abstract. In this paper, some previously implemented notions on robot navigation, collision avoidance and simultaneous target interception are examined and re-implemented on a decentralised autonomous robot group. The key contribution here is a hybrid control approach with emphasis on minimalist communication and no a priori information for negotiating scenarios such as interception, obstacle avoidance and maintaining desired separation among groups. An algorithmic implementation of the hybrid control is presented in this paper. Also, we extend the ideas from our previous work, namely, the use of Kripke Models to represent a system in an intuitive, yet formal manner and proof-check our proposed control strategies using model checking.

1 Introduction

In modern day applications such as surveillance, mapping, exploration, and transportation of large objects that involve the use of mobile robots, some of the key requirements are robust behaviour, decentralised control, autonomy and fault tolerance. Real world situations entail robots to think and act in the wake of sensor and communication constraints and to deal with unstructured and unexpected environments. These aforementioned issues have been treated in isolation or in combination in mobile robotics literature. For instance, Balch and Arkin [1], implemented the behaviour based approach for path planning and formation control in dynamic outdoor environments. Another approach for path planning and autonomous navigation in a similar scenario was proposed in [2]. Howell and Donald [3] used a sensor equipped mobile robot for mapping and localisation. Fierro et al [4] used the classic follow-the-leader for modelling co-operative formations in robots using hybrid control. Burgard et al [5] used a team of robots to explore unknown environments in minimum time.

In all the literature discussed above, the robots maintain desired bearing, optimum separation as well as stay in formation with other robots. To do so, they
must be capable of (a) path planning and localisation, and, (b) making informed and autonomous decisions based on available information. The key factor is to implement these features in a decentralised manner, which has been achievable, largely due to the technical advances in computing power, robust algorithms for decision making and through the use of minimalist communication. Hybrid control approaches are gaining popularity in this field due to their ability to tie in these distinct needs that dominate system design [6].

In this paper, we extend our previous work and add new functionality to the existing hybrid framework. In Jeyaraman et al [7], we have demonstrated an implementation of decentralised, cooperative control within a robot group, navigating in an enclosure. The model was also formally represented using Kripke models. From this formal representation, temporal logic statements of the system behaviour can be derived and model checking can be performed on the system [8].

The main contribution made in this paper is the formalised modelling and implementation of

– co-operative control of a decentralised robot group with minimalist communication and,
– obstacle detection and avoidance with coordinated target intercept in lieu of minimalist communication and no a priori information about the environment.

The robots themselves are assumed to have very little information about the other robots and virtually no knowledge of their environment. A hybrid control approach is used to tackle the problem, where the robots are modelled using continuous mathematics and the decision making that stems from sensing the environment is a discrete one. In our proposed scenario, the robots are constrained to move within an obstructed enclosure. Here, they must be able to navigate effectively and efficiently towards their target, which happens to be the diagonal opposite of the enclosure. If the robots are spaced apart, they should try to converge on each other and subsequently move towards the target. However, once they are close enough, they ought to maintain a minimum separation between each other at all times. There is no a priori knowledge of the environment except for the robots’ initial relative placements.

The remainder of the paper is organised as follows: Section 2 details the modelling approach. Section 3 describes the system representation using Kripke models. Section 4 discusses the algorithmic implementation. Section 5 treats the model checking problem. Section 6 presents our conclusions.

2 Robot Group Model

Mobile robot navigation and obstacle negotiation is an extensively studied topic. Therefore, it was deemed pragmatic to select some of the ideas from the existing cornucopia of relevant literature in order to implement or extend them to our problem. For instance, as in Egerstedt et al [9], we have chosen to keep several behaviours simultaneously active in the robots. By this, there is an obvious
performance gain accrued by simultaneously trying to avoid the obstacles and reach the goal at the same time. Also, point-to-point robot motion at any given two points along the path is represented as smoothly as possible by using a combination of straight-line and circle manoeuvres from [10], [11] and [12], and has been successfully implemented in Jeyaraman et al [7]. In Beard et al [13], the idea of co-ordinating a group of UAVs in order to have simultaneous arrival at the specified target is attempted. This idea is implemented in this paper, where, the robots attempt to intercept a nearest neighbour en route to their goal. As in Balch and Arkin [1] and in Fierro et al [14], the concept of a detection zone that will enable the robots to keep an optimum separation from either obstacles or from each other while travelling to the target is implemented in this paper.

Consider a group of $N$ robots ($N = 3$ in our case), moving in the $\mathbb{R} \times \mathbb{R}$ plane. The kinematics of the robot can be abstracted to the very familiar unicycle model:

\[
\begin{align*}
\dot{x}_i &= v \cos \theta_i \\
\dot{y}_i &= v \sin \theta_i \\
\dot{\theta}_i &= \omega
\end{align*}
\]

All robots travel with the same linear velocity $v$ such that $v = v_1 = v_2 = \ldots = v_N$. Also, the robots are assumed to have a fixed turning radius, represented as $R_{cur}$ (where $R_{cur} > 0$). From this, the turning speed $\omega$ can be given as

\[|\omega| \leq \frac{v}{R_{cur}}\]

For the purposes of this study, time is assumed to share an isomorphic relation with the ordered set of natural numbers, $\mathbb{N}$. In other words, time is discrete.

The robots are all alike and circular (radius of robot $r_{rob} = 0.15\, m$), travel with a uniform linear velocity of $v = 0.1\, ms^{-1}$ and do not stop anywhere, until they have reached the target. Also, they cannot travel backwards. The turning radius of the robot is fixed at $R_{cur} = 0.5\, m$. The robots are supplied with their target as a range, i.e.,

\[
\begin{align*}
x_i &\in \left[ (X_{end} - r_{rob}), (X_{end} - 2r_{rob}) \right] \\
y_i &\in \left[ (Y_{end} - r_{rob}), (Y_{end} - 2r_{rob}) \right]
\end{align*}
\]

The robot group is allowed to navigate in a static unstructured environment. In this case, the environment is a fenced enclosure ($W \times L$), $5\, m \times 10\, m$ in dimension. A schematic of how the enclosure was initially setup for this study is shown in Fig. 1 (A).

The range of the robots’ sensors will define the detection zone of the robots that will activate “collision-avoidance” or “optimum-separation” behaviours in the robots. In our case, this zone is fixed as,

\[L_D = 1.5 \times r_{rob}\]
In other words, the robot can sense information that lies on or inside a circle of radius $L_D$ around itself. Based on this factor, we shall proceed to define other decision making factors the robots will utilise in their co-operative behaviour.

(i) In the beginning, every robot knows the relative placements of its immediate neighbour as a factor of $L_D$.

Separation Factor, $S_F = KL_D$ such that

$$0 < KL_D \leq \left[ \frac{W}{3} - N \times r_{rob} \right]$$

(ii) The intercept behaviour, where robots attempt to intercept their neighbour while travelling to their goal, is activated by the values taken by $S_F$. This
is achieved by defining a lookup that lists several possible lengths of travel against possible values of $S_F$. This lookup is referred to before deciding if interception can be kept active or abandoned for pursuing the main goal. The values held in the lookup table have been empirically determined and shown in Table 1.

Table 1. Lookup table that determines if interception behaviour needs to be activated.

If $S_F$ falls in any of the ranges specified, interception behaviour is activated for a distance $I_D$. If $S_F$ fails the ranges specified, interception behaviour is abandoned.

<table>
<thead>
<tr>
<th>$S_F$</th>
<th>Interception distance ($I_D$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>3$L_D$</td>
<td>4$L_D$</td>
</tr>
<tr>
<td>4$L_D$</td>
<td>5$L_D$</td>
</tr>
<tr>
<td>5$L_D$</td>
<td>6$L_D$</td>
</tr>
<tr>
<td>6$L_D$</td>
<td>7$L_D$</td>
</tr>
<tr>
<td>7$L_D$</td>
<td>8$L_D$</td>
</tr>
<tr>
<td>8$L_D$</td>
<td>9$L_D$</td>
</tr>
</tbody>
</table>

As in [7], the motion constraints imposed on the robot for an observable time interval holds true in this case as well. In [10], it has been shown that in any observable time interval with constrained motion parameters,

(a) Paths that begin with a maximum turning manoeuvre and end with a straight line will be the longest and,
(b) Paths that begin with a straight line manoeuvre and end with a maximum turning manoeuvre will be the shortest.

We shall make optimum use of this result as a switching hybrid control in our path planning algorithms. Owing to the use of a combination of arcs and straight lines the generated trajectory will be smooth as well as efficient thereby overcoming the drawbacks encountered in conventional strategies with switching control.

A robot can optimise its distance travelled, if it can travel in a straight line in the direction of the target, as far as possible. In other words, if a robot with bearing $(x_i, y_i, \theta_i)$ maintains its heading angle at

$$\theta_{opt_i} = \tan^{-1}\left(\frac{Y_{end} - y_i}{X_{end} - x_i}\right)$$

(8)

it will reach the target in the shortest possible path. The robots cannot travel backwards or stop moving at any time during their travel towards the goal. Every time the robot performs an avoidance manoeuvre, the new heading is calculated and if it alters from the optimum angle given by Eqn. 8, the value of
$\omega$ is controlled accordingly (see Fig. 1 (C)) to bring the robot as close as possible to the target.

Consider any two robots designated $i$ and $j$ respectively, and moving towards the goal. Let $(x_{i1}, y_{i1}, \theta_{i1})$ and $(x_{j1}, y_{j1}, \theta_{j1})$ be the current positions of robots $i$ and $j$, respectively at time $T$. If $(x_{i2}, y_{i2}, \theta_{i2})$ and $(x_{j2}, y_{j2}, \theta_{j2})$ are the position and bearing estimates for the robots at time $T + \Delta T$, then the value of $\omega$ of the robots needs to be controlled such that their distance of separation, always lies between $L_D$ or $1.5L_D$. This can be represented as

$$d_{sep} = \sqrt{(x_{i2} - x_{j2})^2 + (y_{i2} - y_{j2})^2},$$

where, $L_D \leq d_{sep} \leq 1.5L_D$

must be satisfied whenever two robots are close enough to sense each other. However, this behaviour can be activated only after the robots are sufficiently close to one other. In [7], this problem was alleviated by placing the robots closer; In this paper, we have extended this approach to include interception as a prerequisite for optimum separation.

2.1 Achieving coordinated time-over-target (TOT)

Some real-life multi-robot applications rely on their effectiveness by assigning more than one robot to converge at a target and with all of them converging at the same time. In such cases, a priori information in the form of an environment map or of the possible threats is provided to the robots [13], [15]. The challenge, however, is to be able to achieve this in an unstructured environment. Our approach is explained below.

As the robots begin to move towards their target, they are provided with the following information about their environment:

- The relative placement of their immediate neighbour, i.e., $S_F$, and,
- The lookup table from Table 1 containing the equivalent values of $I_D$.

From this available information, each robot constructs an imaginary triangle as shown in Fig. 2.

Here, a straight line between the two robots forms the base, while $I_D$ that corresponds to the range of $S_F$ forms one side thereby giving us the estimated distance to intercept, $d_{est}$ as the unknown to be determined. Any new location $(x_b, y_b)$ that is $d$ units away from the current location $(x_a, y_a)$, can be expressed with respect to its current location using the general expression,

$$\begin{bmatrix} x_b \\ y_b \end{bmatrix} = \begin{bmatrix} x_a \\ y_a \end{bmatrix} + d \begin{bmatrix} \cos \theta \\ \sin \theta \end{bmatrix}$$

(10)

where all the symbols retain their usual meaning. The estimated co-ordinates at each of the points $X_2, I$ are determined using Eqn. 10 and $d_{est}$ is determined using the distance formula.
Fig. 2. Representation of an interception triangle of robot $X_1$ against $X_2$

\[ S_F = \text{provided} \]
\[ I_D = \text{looked up from table} \]
\[ d_{est} = \sqrt{(x_I - x_{X_1})^2 + (y_I - y_{X_2})^2} \]

From this information, the necessary heading in order to arrive at the estimated intercept is given by,

\[ \alpha_1 = \cos^{-1} \left( \frac{d_{est}^2 + S_F^2 - I_D^2}{2d_{est}S_F} \right) \]  \hspace{1cm} (11)
As in previous cases, a straight line path is given by the heading angle. The deviation needed in order to attempt an intercept will determine if an interception subgoal will be assigned to that robot. In other words,

\[
subGoal[i] = \begin{cases} 
  \text{true} & \text{if } \alpha_1 \leq \theta_{opt}^i \\
  \text{false} & \text{otherwise}
\end{cases}
\] (12)

This routine is repeated each time after a subgoal has been achieved. When the lookup table no longer returns valid values, there can be no more subgoals. At this point, multiple sets of behaviours namely, optimum separation and move to target is activated and \( \omega \) is controlled accordingly.

3 Formal Model of Scenario

3.1 Introduction to Kripke models and temporal logic

![Kripke model representation](image)

Fig. 3. A Kripke model representation

Temporal logic evolved as a formalism for specifying and verifying the correctness of non-terminating, concurrent computer software (reactive programs) like operating systems, robotic modules, avionics, software controlling mechanical or chemical processes, etc. A reactive program is one that constantly maintains an interaction with its environment and is influenced by it. Therefore, its specification must be done in terms of its ongoing behaviour (that changes with time) [8], [16]. Moor and Davoren [17], apply modal logic in robust controller synthesis for hybrid systems. Tabuada and Pappas [18] also demonstrate the feasibility of LTL for representing discrete time, controllable, linear systems. State transition graphs are well suited for capturing this idea about reactive systems and their behaviour. Temporal logic is a formalism for describing sequences of transitions between states in a reactive system. We have chosen Kripke models of “possible worlds” to represent this idea of a reactive system. Graphically, a Kripke structure can be viewed as a directed graph, i.e., a set of labelled nodes, connected by directed edges (see Fig. 3). Mathematically, a Kripke model, \( \mathcal{M} \), is a triple,

\[
\mathcal{M} = (W, R, L), \quad \text{where}
\]
$W$ is the set of possible worlds,
$R$ is a relation on $W$, $(R \subseteq W \times W)$, accessibility relation,
$L$ is a function $L : W \rightarrow \mathcal{P}(\text{Atoms})$, labelling function.

Kripke models differ from finite automata in the following aspects:

- Only the states in a Kripke model are labelled and not the transitions. A Kripke model is a directed graph unlike automata, which are labelled and directed graphs.
- In a Kripke model all the states are accepting, unlike automata which are defined by a finite set of initial and final states.
- A Kripke model is represented by a set of all possible infinite sequences of states defined by its graph when traversed forever whereas, a finite state automaton is fully defined by a finite table of state transitions.
- Kripke models are always categorised as non-deterministic finite state machines, unlike automata, which are generally bracketed as deterministic.
- A Kripke model directly corresponds to an $\omega$-regular automaton. A Linear Temporal Logic (LTL) formula can be translated into an $\omega$-automaton. This gives an alternative model checking algorithm, that can be performed on-the-fly, for LTL [8].

### 3.2 Kripke model for scenario

We extend the subsumed Kripke model proposed in [7] and add other components that we have proposed in this paper. As mentioned previously, a Kripke model is a directed graph with transitions determined by the accessibility relations. Table 2 tabulates the transition functions as pairs, corresponding to the worlds that they “link”.

**Table 2.** Possible worlds and accessibility relations for the Kripke model from Fig.4

<table>
<thead>
<tr>
<th>Possible Worlds</th>
<th>Accessibility Relations</th>
<th>Possible Worlds</th>
<th>Accessibility Relations</th>
</tr>
</thead>
<tbody>
<tr>
<td>$W_1 - W_2$</td>
<td>$r_1, r_2$</td>
<td>$W_4 - W_3$</td>
<td>$r_7, r_8$</td>
</tr>
<tr>
<td>$W_2 - W_3$</td>
<td>$r_3, r_4$</td>
<td>$W_1 - W_5$</td>
<td>$r_9, r_{10}$</td>
</tr>
<tr>
<td>$W_1 - W_4$</td>
<td>$r_5, r_6$</td>
<td>$W_5 - W_3$</td>
<td>$r_{11}, r_{12}$</td>
</tr>
</tbody>
</table>

- ($W_1 - W_2$): The transition function responsible for switching between these two worlds is based on the factors that initiate interception behaviour in the robots:

$$r_1 = \begin{cases} 
true & \text{if } (I_D \in \text{Table 1 and } \alpha \leq \theta_{opt}) \\
false & \text{otherwise}
\end{cases}$$

(13)
4 Algorithmic Implementation

The aforementioned ideas mentioned in Section 2 were implemented using gcc 2.95.3 C compiler on a 1.8 GHz Intel Pentium IV PC with 512 MB RAM and running SuSE LINUX 8.0. A modular and hierarchical approach was taken in designing the program and implementing it. We list some of the major contributions that should arise from our implementation:
Fig. 5. The robot paths generated for various $S_F$ values are shown. In (a), robot goals were re-assigned to all the robots and robots’ response can be seen. In (b), the separation distance is increased; notice robot 1 performing interception manoeuvre to meet robot 2, while robot 3 has to re-manoeuvre in order to avoid the obstacle. In (c), robot 1 attempts to intercept, but fails to do so. It eventually re-aligns its bearing to reach target. In this case, only two robots have their goals re-assigned and the third one avoids the obstacle and reaches the target. In (d), the same scenario as (c) repeats. The obstacle avoidance is always dynamic and the robots’ response to goal re-assignment can be seen in the results each time.
(i) Co-operative control, i.e., optimum separation, obstacle avoidance and navigation towards the target.

(ii) Interception of the immediate neighbour for coordinated arrival over target.

We wish to re-iterate here, the important ideas and contributions that were proposed for this implementation.

(a) Unlike past approaches listed, the robots have no a priori information about the environment, except the relative placement of its immediate neighbour. Sensor range limits as well as guides robot motion and all decisions are dynamically made and online.

(b) The interception algorithm uses simple trigonometric concepts in order to base its intercept co-ordinates. The approach is computationally non-intensive and involves virtually no communication overhead.

(c) The robots operate with a fixed velocity and utilises the results from [10], implemented in a novel way in [7] and in this paper. This ensures that the switching control is smooth, unlike past attempts in this area.

We have shown in Jeyaraman et al [7] that it is possible to control the dynamics of the system in a fairly accurate way by restricting the kinematics. The algorithms were implemented in a modular fashion so that the robots’ behaviour is truly homogeneous. The algorithms terminated in roughly 6-10 seconds for each run. We analyse some of the results obtained in the following sections:

### 4.1 Separation Factor, $S_F = 2L_D$ 

The results of this run are shown in Fig. 5 (a). Here, the robot’s separation was not large enough to activate interception behaviour. Instead, the group decided to follow the optimum separation and kept in some sort of loose formation. The goal for the robots were re-planned midway in order to assess the response of the robots. As seen in the results, all the robots re-planned their target objective and bearings in order to reach the newly assigned goal.

### 4.2 Separation Factor, $S_F = 4L_D$ 

The results of this run are shown in Fig. 5 (b). Here, the separation is enough to initiate interception behaviour. This is demonstrated by Robot 1 which initiates repeated combination manoeuvres to catch up with its neighbour. It can be seen that it uses the “arc first” manoeuvre effectively in order to achieve interception. Once again, the goals for the robots were re-planned midway in order to assess the response of the robots. As seen in the results, all the robots re-planned their target objective and bearings in order to reach the newly assigned goal.

### 4.3 Separation Factor, $S_F = 6L_D$ 

The results of this run are shown in Fig. 5 (c). Here, the separation is enough to initiate interception behaviour. In this case, interception does not occur as separation is higher. However, the paths taken by robots 1 and 2 ensure that
they have co-ordinated arrival over target. Once again, the goal for the first two robots were re-planned midway in order to assess the response of the robots. Robot 3, whose goal was not re-planned, reached the target, while the others stayed true to their goal positions.

4.4 Separation Factor, \(S_F = 8L_D\)

The results of this run are shown in Fig. 5(d). As in the previous case, interception could not be achieved because of the large distances, although the attempt has once again kept the robots in a position to achieve co-ordinated TOT. As in the previous case, re-planning was implemented on only two robots, as can be seen.

5 Model Checking

Due to lack of space, we have chosen not to include model checking results here. The final paper will carry the model checking results.

6 Conclusions

We have proposed a novel hybrid control scheme for decentralised, co-operative control with minimalist communication and no \textit{a priori} information for a group of robots. To this effect we have demonstrated algorithmically several features such as online path planning from sensed information, dynamic re-planning of goals (Fig. 5(a)-(d)), interception of neighbours to facilitate co-ordinated intercept and to some extent, dynamic obstacle avoidance. We have been able to show that, the solution provided by the interception algorithm is locally optimal, i.e., at least two robots out of the three converge to a formation and co-operatively move towards the goal, i.e., co-ordinated TOT is achieved in the end. We hope to improve this behaviour by incorporating some form of feedback or communication that will exist between the robots so that this can be achieved. We have also demonstrated how fixed velocity systems can still be used effectively in scenarios that involve variable velocities. In the future, we shall investigate performance gain by incorporating variable velocities in the robots. Also, we have proposed the use of Kripke models of “possible worlds” to formally represent our system, in order to be able to predict and facilitate performance. As it is manually impossible to predict the number of states the system would reach before termination, such a compact representation will greatly aid us in understanding complex systems’ behaviour and in predicting their performance.

Acknowledgements

The first author wishes to thank Satheesh Jeyaraman for his input and constructive suggestions on the algorithms and Darina Fišerová for her help in preparation of the manuscript.
References

