



Discovering Emergent Agent Behaviour with Evolutionary Finite State Machines

Martin Masek¹ , Chiou Peng Lam¹ , Lyndon Benke² , Luke Kelly¹ ,
and Michael Papisimeon² 

¹ School of Science, Edith Cowan University, Perth, Australia

² Defence Science and Technology Group, Melbourne, Australia
{lyndon.benke,michael.papasimeon}@dst.defence.gov.au

Abstract. In this paper we introduce a novel approach to discovering emergent behaviour in multiagent simulations, using evolutionary finite state machines to model intelligent agents in an adversarial two-player game. Agent behaviour is modelled as a finite set of predetermined states. The logic that leads to transitions between states is evolved to maximise fitness, which is determined through execution in a constructive simulation environment. The resultant evolved finite state machine (E-FSM) is evaluated for two finite state machine implementations, one with states specifically designed to perform a known behaviour and the other with states consisting of generic actions. Our experiments demonstrate that this approach can discover complex emergent behaviours from simple, generic actions, and use these behaviours to achieve a position of tactical superiority in the domain of air combat simulation.

Keywords: Emergent behavior · Evolutionary algorithms
Multiagent simulation · Air combat simulation

1 Introduction

Increasingly, the military uses simulations for defence applications, such as training, concept development and experimentation, and as an alternative to live training exercises which involve expensive aircraft and require the presence of highly trained pilots. These simulations have incorporated intelligent agents to model individual and team decision-making, for a variety of reasons including the development and assessment of tactics for air combat [2]. A typical agent has a role, such as ally or adversary, and its behaviour is traditionally scripted by hand mapping observations to actions.

The process of scripting these behaviour models for specific simulation environments is labour intensive, costly, requires domain expertise, and may require the use of dedicated agent behaviour authoring tools. Any subsequent variations to the agent behaviour require manual modifications to the existing model. In

This work was supported by the Australian Defence Science and Technology Group.

addition, these scripted behaviour models are limited in their ability to discover novel behaviours.

Adaptive machine learning systems have been employed as an alternative to human test pilots and live training exercises [13]. Smith et al. [13] argued that this approach has numerous advantages, namely, a distilled analytical model that captures combat simulation, eliminates bias from the pilots' previous experiences, and does not have the constraints typically associated with real-time simulations.

Technological advances for fighter aircraft, such as stealth, advanced avionics and weapons, electronic warfare, and increased networking capabilities, require sophisticated tactics to be exploited effectively. As a result, there is a need for new techniques to adequately explore fighter combat behaviour. Artificial intelligence techniques, such as automated planning [8,9] and differential game theory [7], have been employed to automate the discovery of new tactical behaviours by optimising action selection in a close range two player air combat scenario.

In this paper, we demonstrate that the integration of evolutionary algorithms with finite state machines provides a viable approach to tactical behaviour discovery. The approach involves the evolution of a behavioural model in the form of a finite state machine (FSM) using genetic algorithms, to produce an Evolved Finite State Machine (E-FSM) that can automate the generation of complex tactical behaviours in multiagent simulations.

The contributions are (1) a generalisable evolutionary approach to evolving finite state machine transitions in multiagent simulations, and (2) the development of a chromosome representation for evolving rules for transitions of finite state machines.

The paper is structured as follows: A discussion of related works is first presented in Sect. 2. Section 3 describes the air combat problem, namely the stern conversion manoeuvre, and the corresponding stern conversion agent controller used in this study. Our evolutionary approach is outlined in Sect. 4, followed descriptions of the sets of experiments and associated results and discussion in Sect. 5. Lastly, conclusions and future works are presented in Sect. 6.

2 Related Work

Existing research has explored the use of machine learning to automatically generate behaviour models incorporating techniques such as evolutionary algorithms and dynamic scripting, in conjunction with employing FSMs or behaviour trees as models for capturing agent behaviour.

The Smart Bandits project [11] employed machine learning techniques to generate FSM-based behaviour models that can be used for air-to-air tactic training. While Smart Bandits provides no adaptive capabilities, Toubman et al. [14] incorporated dynamic scripting in their approach for developing adaptive FSMs that can be used against opponents in air combat.

Revello [10] presented a GA-based approach for generating war game strategies whereby the various ship groups demonstrated emergent behaviours, moving

in a coordinated fashion even though there was no exchange of information in terms of their respective movements.

Smith et al. [13] evolved a learning classifier system that produced novel one-vs-one WVR manoeuvres in the role of a fighter pilot using the AASPEM system as the simulator environment. The GA population consists a set of classifiers and through an iterative process of fitness evaluation of the classifiers, selection and genetic operations, produced several novel strategies that were subsequently evaluated and approved by test fighter pilots.

Other researchers such as Mulgund et al. [5] and Keshi et al. [4] have applied genetic algorithms to optimise tactical parameters in BVR involving scenarios of many-vs-many using hierarchical encoding involving binary codes for chromosome representation. Yao et al. [15] described an approach where air combat manoeuvres are represented using behaviour trees and through grammatical evolution, generates adaptive human behaviour models for BVR engagements.

While Toubman et al. [14] and others have applied evolutionary algorithms to FSM behaviour models, those approaches relied on hand-authored tactical behaviours. The approach presented here obviates the need for problem-specific tactics, by evolving transitions between generic agent actions using kinematic properties of the entities. This enables the emergence of complex behaviours without predisposing the system towards known solutions. Unlike the approach taken by researchers such as Smith et al. [13], by incorporating an FSM our approach produces human-readable tactics, and allows the modelling of agent behaviours with higher levels of complexity if required, enabling the discovery of emergent behaviours for problem scenarios with different levels of abstraction.

3 Problem Description

The problem used to evaluate our approach is that of an aircraft that aims to manoeuvre into a particular position of tactical superiority with respect to a single adversary aircraft. This position is defined as being behind the adversary and following it. In this position, the aircraft can fire on its adversary, whilst being out of range of the adversary's weapons.

The classical scenario that models this is the stern conversion intercept (described by Shaw [12]), a two-player scenario where the aircraft are within visual range and initially flying towards each other (Fig. 1). Execution of the manoeuvre as described relies on a number of tactical parameters, which are dependent on the manoeuvring capabilities of the aircraft.

An implementation of the stern conversion manoeuvre as a finite state machine is shown in Fig. 2, where each state represents the execution logic of a single section or subtask of the manoeuvre. The transitions between the states, based on various conditions defined by Shaw [12], result in a sequence that realises the complex behaviour. A description of the states is given in Table 1.

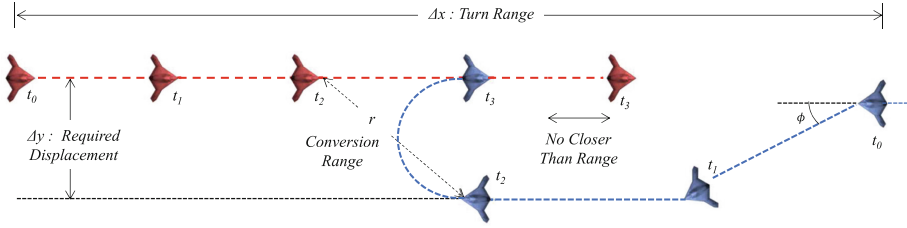


Fig. 1. The sequence of subtasks and tactical parameters required to execute the classical stern conversion manoeuvre.

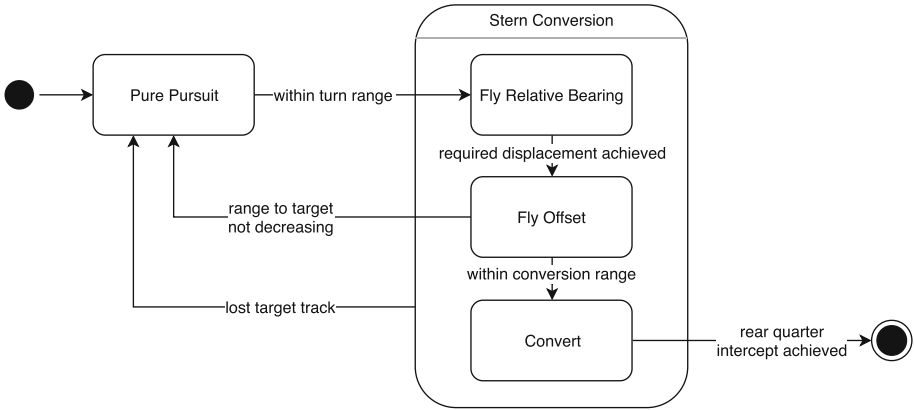


Fig. 2. FSM implementing the Shaw stern conversion manoeuvre.

4 Proposed Approach

4.1 A Flexible FSM Model

Our approach to the exploration of air combat strategies employs a flexible FSM model, where a set of states is provided as input and an evolutionary algorithm is used to determine appropriate transition conditions so that the FSM can act as an agent controller that achieves a particular goal. The transition conditions in our approach are based on the kinematic properties of an aircraft and a measure of goal achievement for the evolutionary algorithm is obtained through agent-based simulation.

A generic FSM controller, employing n states, is shown in Fig. 3. From each state it is possible to transition into any other state, depending on a set of conditions. Each state corresponds to an action that is performed in that state, such as flying an aircraft in a particular manner, and a set of conditions that are constantly checked while in that state to determine if the FSM should transition to another state.

The functional logic of the generic FSM is provided in Algorithm 1, where the FSM starts in State 1. In the case where transition conditions for moving

Table 1. States and transitions for the FSM stern conversion agent

State	Action	Transitions
(a) Pure pursuit	Point aircraft at and fly directly towards red aircraft	If distance to red aircraft less than turn range and red aligned with blue, transition to (b)
(b) Fly relative bearing	Turn by turn angle and fly straight on that bearing	If lateral separation between red and blue greater than required displacement transition to (c)
(c) Fly offset	Fly parallel to heading of red aircraft	If distance between red and blue less than conversion range transition to (d). If distance between red and blue not decreasing transition to (a)
(d) Convert	Turn to match red heading. Approach no closer than no closer range	If the rear quarter intercept criteria have been met then the maneuver reaches the end/final state as it has been successfully executed

into more than one state become valid, the next state is chosen probabilistically. In its simplest form, the choice of valid transition can be random, as was done in our experiments, or each transition can be assigned its own probability.

The key difference between the approach taken here is and a regular finite state machine is that in this case we only assume what states the pilot agent can be in. We don't make any assumptions about the transition events or the transition probabilities between states. Rather, the transitions between the pre-determined states are evolved and hence generated dynamically. This allows for the possibility of new tactical behaviour to emerge and to be potentially discovered.

Algorithm 1. Functional logic for the generic FSM agent controller.

```

currentState = State1; nextPossibleStatesList = empty;
loop
  performAction(currentState);
  for nextStateToCheck ← 1 to n do
    if transitionConditionsSatisfied(currentState, nextStateToCheck) then
      | append nextStateToCheck to nextPossibleStatesList;
    end
  end
end
if nextPossibleStatesList not empty then
  | currentState = randomChoice(nextPossibleStatesList);
  | nextPossibleStatesList = empty;
end
end
end

```

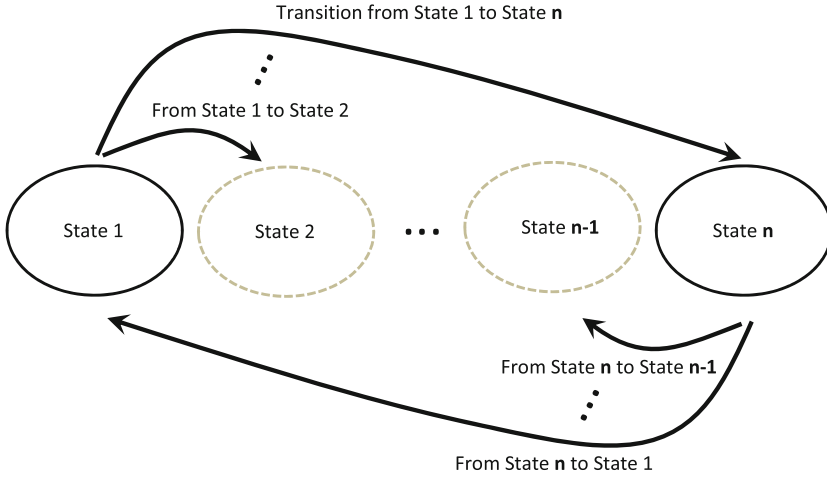


Fig. 3. A generic n -state finite state machine where each state corresponds to some action taken by the agent. From each state, a transition can occur to any other state. The conditions that would lead to state transitions are evolved in our approach to suit a particular task.

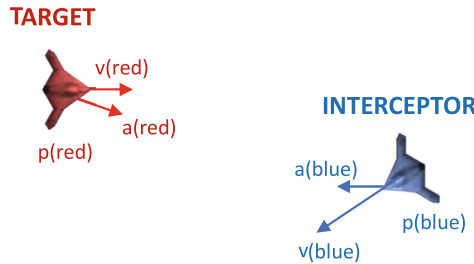


Fig. 4. Kinematic properties of the Red and Blue aircraft; each has a position (\mathbf{p}), velocity vector (\mathbf{v}) and acceleration vector (\mathbf{a}). (Color figure online)

4.2 Kinematic Transition Model

We base the transition conditions for the air combat domain on the kinematic properties of the aircraft taking part in the scenario: position (\mathbf{p}), velocity (\mathbf{v}) and acceleration (\mathbf{a}). For the case of two aircraft (denoted Blue and Red), these parameters are shown in Fig. 4. Each kinematic parameter is a vector quantity with three Cartesian components, along the x , y and z axes.

The kinematic properties are transformed from values with reference to a fixed world coordinate system, to a coordinate system relative to the aircraft controlled by the FSM. This ensures that the transition parameters are rotation and translation invariant (i.e. not specific to a particular position and orientation in the world), and is obtained by calculating the difference vectors for each parameter:

$$\begin{aligned}
 \Delta \mathbf{p} &= (\Delta p_x, \Delta p_y, \Delta p_z) = (p_{xRed} - p_{xBlue}, p_{yRed} - p_{yBlue}, p_{zRed} - p_{zBlue}) \\
 \Delta \mathbf{v} &= (\Delta v_x, \Delta v_y, \Delta v_z) = (v_{xRed} - v_{xBlue}, v_{yRed} - v_{yBlue}, v_{zRed} - v_{zBlue}) \\
 \Delta \mathbf{a} &= (\Delta a_x, \Delta a_y, \Delta a_z) = (a_{xRed} - a_{xBlue}, a_{yRed} - a_{yBlue}, a_{zRed} - a_{zBlue})
 \end{aligned}$$

where $\Delta \mathbf{p}$ is the distance vector between the two aircraft, and $\Delta \mathbf{v}$ and $\Delta \mathbf{a}$ are the difference vectors between the velocity and acceleration vectors. The three vectors are then rotated to compensate for the heading direction of the blue aircraft. After transformation the values correspond to the position, velocity and acceleration of the red aircraft as they would be perceived by the pilot of the blue aircraft. The values of these transformed kinematic parameters can then be used to determine whether a transition should take place.

A wide number of algorithmic approaches can be employed to implement the transition conditions. We employ a simple model, checking that $\Delta \mathbf{p}$, $\Delta \mathbf{v}$ and $\Delta \mathbf{a}$ are within a certain range of values. The minimum and maximum bounds on the transition condition ranges are represented by a set of constants, determined through an evolutionary approach (described in Sect. 4.3). Evaluation for the transition from a particular state i to another state j requires 18 constants since there are three kinematic parameters ($\Delta \mathbf{p}$, $\Delta \mathbf{v}$ and $\Delta \mathbf{a}$), each of these has three Cartesian components (x , y and z), and each of these has both a lower and upper bound. For an FSM with n states, in each state a total of $n - 1$ transitions need to be checked, thus $18 \times (n - 1)$ boundary checks, making the computational complexity of the FSM transition model $O(n)$. The storage complexity is $O(n^2)$ as the total number of constants to be stored is:

$$n_{constants} = 18 \times (n_{states} - 1) \times n_{states} \quad (1)$$

These constants are denoted by $A_{ij}, B_{ij}, C_{ij}, D_{ij} \dots R_{ij}$. To determine if a transition from current state i to next state j can be taken, the current $\Delta \mathbf{p}$, $\Delta \mathbf{v}$ and $\Delta \mathbf{a}$ is evaluated against the bounds, as shown in Table 2.

Table 2. An example of the transition logic conditions that must all be satisfied for a change of state from state i to state j . The 18 transition parameters (A to R) act as thresholds for the relative position ($\Delta \mathbf{p}$), velocity ($\Delta \mathbf{v}$) and acceleration ($\Delta \mathbf{a}$) in each transition.

Relative position	Relative velocity	Relative acceleration
$A_{ij} < \Delta p_x < B_{ij}$	$G_{ij} < \Delta v_x < H_{ij}$	$M_{ij} < \Delta a_x < N_{ij}$
$C_{ij} < \Delta p_y < D_{ij}$	$I_{ij} < \Delta v_y < J_{ij}$	$O_{ij} < \Delta a_y < P_{ij}$
$E_{ij} < \Delta p_z < F_{ij}$	$K_{ij} < \Delta v_z < L_{ij}$	$Q_{ij} < \Delta a_z < R_{ij}$

4.3 Genetic-Based Approach to Transition Evolution

Our evolutionary approach to determining the optimal boundaries for transition conditions is based on the genetic algorithm (GA), originally developed by Holland (1992) [3]. The GA takes an initial set of candidate solutions to the problem,

called the *population*. For the purposes of the algorithm, individuals are encoded as a set of attributes, the *gene*, with the set of genes called a *chromosome*.

In our approach, the boundary constants each form a gene, with the complete set of constants forming the chromosome. Thus for an FSM with n states, given Eq. 1, there are $18 \times (n - 1) \times n$ genes in each chromosome. The chromosome representation is shown in Fig. 5, with the constants laid out sequentially. For the purposes of the GA, each constant is stored as a real number, normalised to be in the range $[0.0-1.0]$. The values are mapped during execution to an actual range for the kinematic parameters by considering their physical bounds, based on mission parameters and aircraft capabilities.

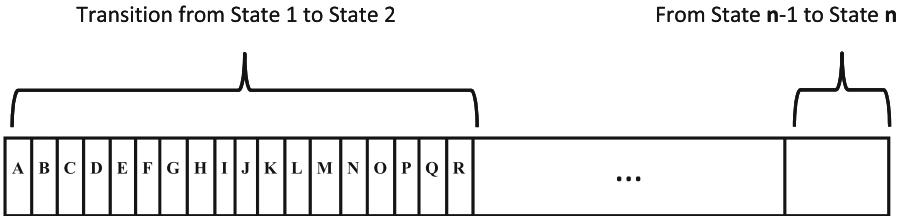


Fig. 5. Chromosome representation for evolving FSM transitions.

5 Evaluation

5.1 Experimental Results

The evolved finite state machine (E-FSM) approach described in Sect. 4 was evaluated for two FSM implementations (described below), one with states specifically designed to perform a known behaviour, and the other with states consisting of generic actions utilised by Park et al. [7]. The scenario used for our experiments is a two player close-range air combat engagement, as per the initial conditions for Shaw’s stern conversion scenario described in Sect. 3.

Each implementation of the E-FSM approach is evolved against a set of four opponent models, described in Table 3. The range of permitted transition boundary parameter values is given in Table 4, chosen based on the scenario scale and aircraft characteristics to speed up convergence by avoiding values that in practice could not be reached. Each experiment was repeated 30 times with different initial populations, to examine the range of solutions produced by the non-deterministic evolution process.

Genetic Algorithm Parameters. The population is initialised by generating a set of individuals such that each gene that corresponds to a minimum bound has its value set to a random number in the range of $[0, 0.5]$, while each gene

Table 3. Description of the four opponent models used to evaluate each implementation of the E-FSM approach.

Opponent	Behaviour description
Straight Line	Non-reactive; flies in a straight line as per the red target aircraft in Shaw’s stern conversion (Fig. 1)
Pure Pursuit	Pursues the blue fighter using the <i>pure pursuit</i> behaviour described in Table 1(a)
Shaw	Attempts the classical stern conversion manoeuvre using the stern conversion specific FSM described in Table 1, with hand-selected transition parameters
Evolved Shaw	Attempts the stern conversion manoeuvre using the stern conversion specific FSM described in Table 1, with transition parameters evolved against a straight line agent

Table 4. Kinematic parameter ranges used to reduce experimental run time.

Kinematic parameter	Encoded range	Actual range
Δp	[0, 1]	[-48152, 48152] m
Δv	[0, 1]	[-1000, 1000] ms ⁻¹
Δa	[0, 1]	[-100, 100] ms ⁻²

that corresponds to a maximum bound has its gene value initialised to a random number in the range [0.5, 1].

Evolution proceeds for a pre-determined number of generations, each of which involves comparing the performance of individuals in the candidate population through 5 runs in a constructive simulation environment, with a maximum simulation run-time of 250 s.

The population is updated after each generation by copying a number of individuals, selected using Stochastic Universal Sampling [1], and the single most fit candidate (the *elite*), into the next generation. Selected individuals are combined using single point crossover and two point crossover [3]. Mutation is controlled using the Gaussian mutation operator, as it is flexible enough to allow for both fine tuning of solutions and searching of the domain. The value for a mutated gene is calculated using the equation $x = x + \mathcal{N}(0, 1)$ where $\mathcal{N}(0, 1)$ is the Gaussian Normal distribution with a mean of 0 and a standard deviation of 1. The probability as to whether a gene undergoes mutation is associated with the mutation probability, p_m , and this has been assigned a value of 0.1, based on initial experimental results.

Experiments are terminated when either of the following conditions are met: the number of generations reaches a pre-defined maximum number, or there has been no improvement in the fitness value in the population for N consecutive generations.

Fitness Evaluation. In this study, the fitness of an individual is calculated from the output of a set of runs in the constructive air combat simulator ACE Zero [6]. We base our success measure for blue on the achievement of a position of superiority, defined as being behind the red agent and following it. We consider the blue aircraft to have succeeded if during the simulation run the following criteria have been met (illustrated in Fig. 6):

1. Target aircraft is within 30° of attacking aircraft nose ($\phi_1 < 30$)
2. Attacking aircraft is within 30° of threat aircraft tail ($\phi_2 < 30$)
3. Range to the target aircraft is between 500 and 3000 feet ($500 < r < 3000$)
4. Separation in altitude is less than 500 feet ($\Delta a < 500$)
5. Difference in velocity is less than 100 knots ($\Delta v < 100$)

To calculate the fitness of a blue aircraft in a particular simulation run, we initialise its fitness to 0 at the start of the simulation, then check each of the six conditions at intervals of 1s of simulated time. For each condition that is true at the particular point in time we add 1 to the fitness. Thus, the more conditions that are true at a particular time, the higher the fitness for that time interval. Fitness is summed across time intervals, so that a higher fitness results the longer a condition is true.

While more criteria could be considered, for example that the above criteria be met continuously for a long enough duration to launch a weapon, through our experiments we confirmed that the above five criteria were sufficient to produce valid solutions.

One simulation run results in a single fitness score. Due to the non-determinism that is present when multiple transition conditions are satisfied, we take the average fitness of five simulation runs and associate that with the individual.

Problem-Specific E-FSM Results. The problem-specific E-FSM has states hand-coded to enact the stern conversion manoeuvre as described by Shaw [12] (Fig. 2), with the original tactical-parameter-based transitions replaced by the flexible kinematics-based model described in Sect. 4. This FSM has 4 states, resulting in a chromosome with 216 genes as per Eq. 1. An initial population of 50 individuals is evolved through a maximum of 300 generations, and the individual with the highest fitness after termination is selected for examination.

Figure 7 illustrates exemplary results for each of the opponent types described in Table 3. Against the Straight Line opponent, the evolved problem-specific FSM found the classical stern conversion sequence (that is, the states in Table 1 executed in order), despite having no knowledge of the tactical parameters traditionally used to execute the state transitions. Against the Pure Pursuit opponent, the agent learned to stay in the Pure Pursuit state. In the case of the Shaw opponent, a hand-optimised stern conversion FSM, the problem-specific E-FSM learned a novel tactic, waiting for its opponent to turn away before turning to follow it as it passed. Against the more adept Evolved Shaw opponent, which had itself been evolved against a Straight Line agent, the E-FSM learned to

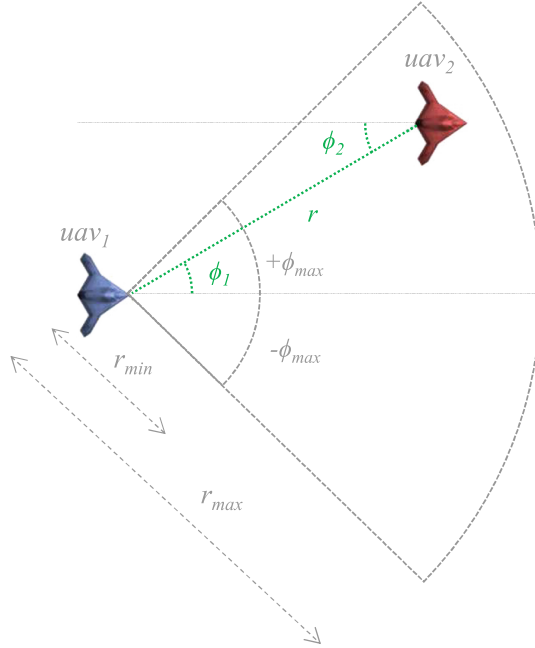


Fig. 6. Illustration of the criteria used to evaluate the fitness of an individual. Blue is in a position of superiority, corresponding to a high fitness score.

counteract the behaviour of its opponent to complete the manoeuvre behind it. In all cases, the evolutionary approach successfully found effective behaviours to achieve a position of tactical superiority.

Generic E-FSM Results. The generic E-FSM has states based on generic aircraft manoeuvre actions, taken from Park [7] (illustrated in Fig. 8). In previous versions of the evolutionary finite machine, the states represented either high level or intermediate level goals or maneuvers that the pilot agent was trying to achieve (such as flying an offset maneuver). In this iteration described here, break the maneuvers down even further into low level actions such as flying left and up, level flight and right and down. These represent some of the lowest level actions a pilot can take to control an aircraft. By assembling a sequence of these low level aircraft control actions, a pilot can assemble a different higher level maneuvers that will enable it to undertake to model basic fighter combat. Through the evolution of the transitions between these low level actions—states we can generate air combat behaviour against a maneuvering opponent without the constraints of a pre-determined tactic such as that described by Shaw [12].

Due to the generic E-FSM having 7 states, it results in a chromosome with 756 genes as per Eq. 1. As the chromosome size is much larger than for the problem-specific E-FSM, resulting in a larger search space, more extensive exploration

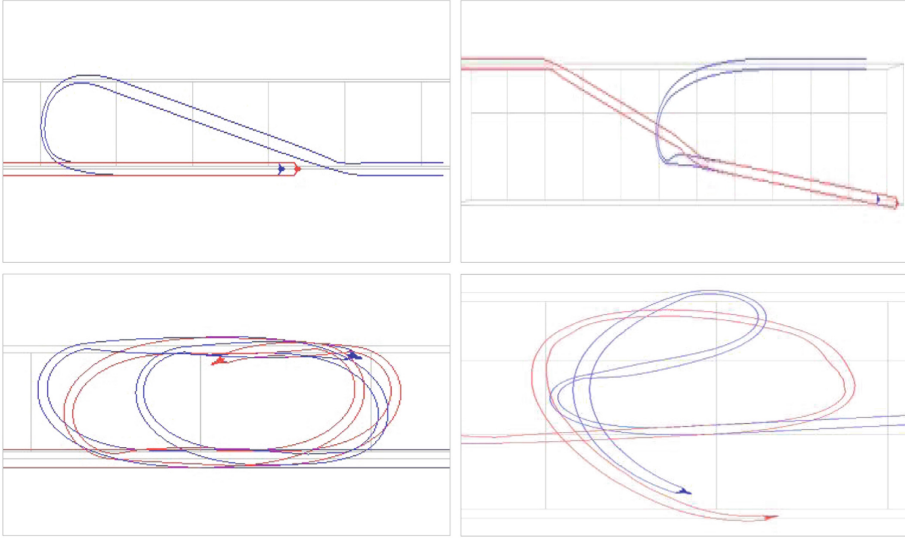


Fig. 7. Example traces for the problem-specific E-FSM against red opponents (clockwise from top left): Straight Line, Shaw, Evolved Shaw, Pure Pursuit. (Color figure online)

was enabled through a larger population of 100, and by increasing the maximum number of generations to 1000. The large size of the chromosome also led us to lower mutation probability to 0.001 and use two-point crossover to prevent excessive modification between generations.

Figure 9 illustrates exemplary results for each of the opponent types described in Table 3. Against the Straight Line opponent, the generic E-FSM found an effective set of transitions between the generic manoeuvre states to approximate the classical stern conversion sequence, although with a lower average fitness score than the problem-specific E-FSM (which has states specifically designed for this opponent). Against the Pure Pursuit opponent, the agent learned a behaviour that approximated the hand-coded pure pursuit behaviour of the problem-specific FSM, continually transitioning between basic manoeuvres to follow its opponent. The pursuit behaviour discovered by the generic E-FSM achieved a higher fitness than the hand-coded Pure Pursuit state, suggesting that the agent had found a superior tactic (most likely the more efficient lead pursuit). In the case of the Shaw opponent, the generic E-FSM learned a similar strategy to the problem-specific E-FSM, waiting for its opponent to turn away before turning to follow it. Against the more adept Evolved Shaw opponent, the generic E-FSM achieved significantly improved performance over the problem-specific E-FSM, discovering a novel tactic (drawing its opponent into a turn before looping around behind it) that was not found when the hand-coded, stern conversion specific states were used. In all cases apart from the straight line opponent, the generic E-FSM attained significantly higher fitness scores than the problem-specific E-FSM.

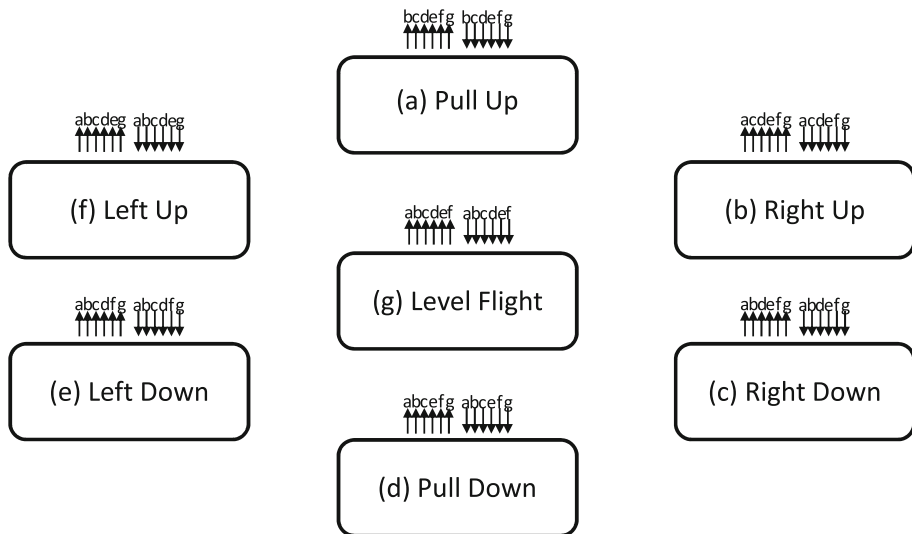


Fig. 8. Representation of the states in the Generic E-FSM, where each state represents a low level aircraft control. The arrows above each state indicate that the E-FSM can evolve to transition to any possible state.

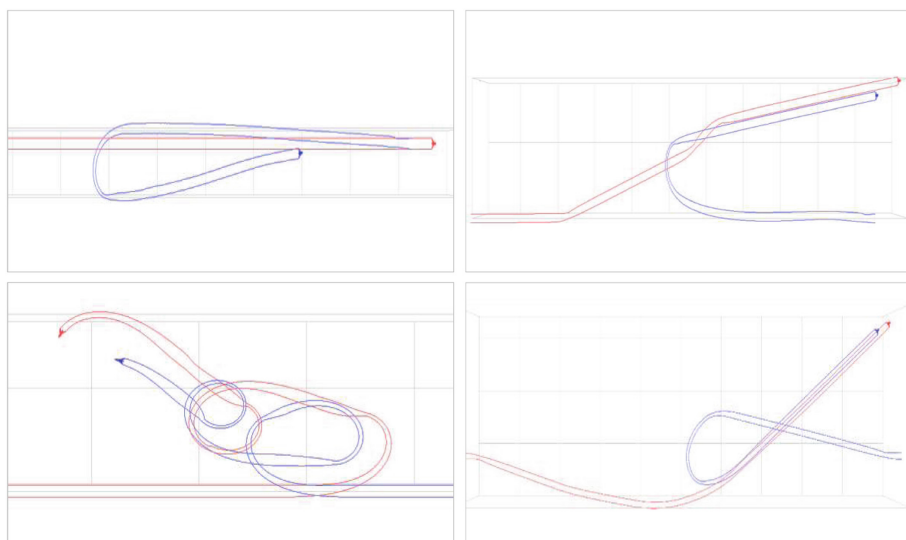


Fig. 9. Example traces for the generic E-FSM against red opponents (clockwise from top left): Straight Line, Shaw, Evolved Shaw, Pure Pursuit. (Color figure online)

5.2 Discussion

In comparison to traditional approaches such as the hand-coded state transition logic described in Sect. 3, where the transitions are specific to the problem and must be determined by the analyst for each combination of aircraft, the kinematic approach was able to discover a sequence of transitions to enact a successful rear-quarter intercept without prior knowledge of the flight characteristics of either aircraft. In addition, the generic E-FSM was able to generate complex emergent behaviours, such as pursuit and drawing the opponent into an advantageous position, from simple aircraft flight control actions.

The experiments highlight a number of interesting properties of using an evolutionary approach to optimise tactical behaviour, primarily the impact of the particular fitness function chosen, and the relationship between the level of complexity of the FSM states and the novelty of discovered behaviours.

Impact of the Fitness Function. The fitness function used in our experiments was based purely on the desired final outcome (Fig. 6), calculated by aggregating points associated with five criteria at 1 s intervals and summed over a period of 250 s of simulation time. Since the objective is to maximise the fitness function, the evolved model is naturally biased towards solutions that avoid any intermediate manoeuvres that reduce fitness, although these may subsequently be helpful in better achieving the final goal. For example, when evolving the problem-specific E-FSM against a straight line opponent, the classical stern conversion manoeuvre (Fig. 1) achieves a rear-quarter intercept faster than the greedy behaviour of remaining in the Pure Pursuit state. However, the stern conversion manoeuvre begins with a turn away from the opponent, resulting in an initial loss of fitness. In comparison, a solution where the blue aircraft points continuously at its opponent over the same time period (pure pursuit) will be ranked higher in fitness initially, despite ultimately taking longer to achieve the rear-quarter intercept. As a result, it is important to develop the fitness function carefully to avoid biasing discovered behaviours in this way, and to allow sufficient evolution time for manoeuvres with lower initial fitness to be found.

Transitional Complexity of Solutions. A factor in finding an effective set of transition conditions for the FSM is the number of transitions that need to be made to reach an optimal solution. The evolutionary process favours solutions with lower transitional complexity. For example, performing the prescribed stern conversion manoeuvre depends on the problem-specific E-FSM executing four states at the right time and in the right order, requiring the correct evolution of up to 216 transition condition parameters. On the other hand, the Pure Pursuit state provides a relatively strong solution on its own. This means that solutions that involve staying in the Pure Pursuit state have a very low transitional complexity, and require the correct optimisation of fewer transition condition parameters (none if the FSM starts in the Pure Pursuit state). As a result, sufficient exploration should be enabled during evolution to search the

solution space, so that more complex manoeuvres can be found. The more complex a manoeuvre, in terms of the number and sequence of transitions that need to occur, the longer it can be expected to take to evolve.

State Complexity and Solution Novelty. It is observed from the experimental results that, when implementing the E-FSM approach, the use of more complex, problem-specific states predisposes the evolutionary search toward known solutions, while the use of simpler states produces more novel solutions (although at a computational cost). While the problem-specific E-FSM often converged to a solution within 20 generations, it did so in many cases by settling quickly on a sub-optimal solution, such as staying in the Pure Pursuit state (which is able to achieve the final goal on its own). In contrast, the states of the generic E-FSM correspond to simple directional changes for the aircraft, so evolving a novel sequence of state transitions is the only way to achieve the target goal without an initial workable solution.

6 Conclusion

We have demonstrated an effective approach to discovering emergent tactical behaviours, using the combination of evolutionary algorithms with finite state machines in the context of adversarial air combat. The incorporation of FSMs produces human-readable tactics, and enables the modelling of agent behaviour at varying levels of complexity, avoiding the predisposition towards known solutions that results from hand-coding behaviours, while enabling the modelling of actions at higher levels of abstraction as required.

It was found that there is a strong relationship between the complexity of the FSM states, the time taken to find an effective solution, and the novelty of discovered behaviours. For example, a generic FSM, with simple states representing low-level aircraft directional changes, took many more generations to find an optimal solution than a problem-specific FSM, whose more complex states were hand-coded for the specific scenario. However, the generic FSM achieved superior performance when evolved against reactive opponents, and discovered novel tactics that were not seen when using problem-specific states.

Future work will involve scaling the approach presented here to more complex scenarios involving teams of agents employing beyond visual range sensors and weapons (requiring more sophisticated transition conditions than the kinematic parameters used to determine state transitions in this study), and the integration of co-evolutionary methods.

References

1. Baker, J.E.: Reducing bias and inefficiency in the selection algorithm. In: Proceedings of the Second International Conference on Genetic Algorithms, vol. 206, pp. 14–21 (1987)

2. Heinze, C., Papasimeon, M., Goss, S., Cross, M., Connell, R.: Simulating fighter pilots, Birkhuser, Basel, pp. 113–130, November 2008. https://doi.org/10.1007/978-3-7643-8571-2_7
3. Holland, J.H.: *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*. MIT Press, Cambridge (1992)
4. Keshi, Z., Zhengping, W.: On optimizing large-scale air-combat formation with simulated-annealing GA (genetic algorithm). In: 24th International Congress of the Aeronautical Sciences (2004)
5. Mulgund, S., Harper, K., Krishnakumar, K., Zacharias, G.: Air combat tactics optimization using stochastic genetic algorithms. In: 1998 IEEE International Conference on Systems, Man, and Cybernetics, vol. 4, pp. 3136–3141. IEEE (1998)
6. Papasimeon, M., Benke, L., Brain, R., Finkelstein, L.: Multiagent simulation of adversarial socio-technical systems. Presented at the Industrial Track, 2018 International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS 2018) (2018)
7. Park, H., Lee, B.Y., Tahk, M.J., Yoo, D.W.: Differential game based air combat maneuver generation using scoring function matrix. *Int. J. Aeronaut. Space Sci.* **17**(2), 204–213 (2016)
8. Ramirez, M., Papasimeon, M., Benke, L., Lipovetzky, N., Miller, T., Pearce, A.R.: Real-time UAV maneuvering via automated planning in simulations. In: International Joint Conferences on Artificial Intelligence Organization, pp. 5243–5245, August 2017. <https://doi.org/10.24963/ijcai.2017/778>, <https://www.ijcai.org/proceedings/2017/778>
9. Ramirez, M., et al.: Integrated hybrid planning and programmed control for real-time UAV maneuvering. In: Proceedings of the Seventeenth International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2018, July 2018
10. Revello, T.E., McCartney, R.: Generating war game strategies using a genetic algorithm. In: Proceedings of the 2002 Congress on Evolutionary Computation CEC 2002, vol. 2, pp. 1086–1091. IEEE (2002)
11. Roessingh, J., et al.: Modelling CGFs for tactical air-to-air combat training motivation-based behaviour and machine learning in a common architecture. Technical report, National Aerospace Laboratory NLR (2011)
12. Shaw, R.L.: *Fighter Combat*. Naval Institute Press, Annapolis (1985)
13. Smith, R., Dike, B., Ravichandran, B., El-Fallah, A., Mehra, K.: Discovering novel fighter combat maneuvers: simulating test pilot creativity. In: *Creative Evolutionary Systems*, pp. 467–VIII. Elsevier (2002)
14. Toubman, A., Roessingh, J.J., Spronck, P., Plaat, A., van den Herik, H.J.: Rapid adaptation of air combat behaviour. In: *ECAI. Frontiers in Artificial Intelligence and Applications*, vol. 285, pp. 1791–1796. IOS Press (2016)
15. Yao, J., Huang, Q., Wang, W.: Adaptive human behavior modeling for air combat simulation. In: 2015 IEEE/ACM 19th International Symposium on Distributed Simulation and Real Time Applications (DS-RT), pp. 100–103. IEEE (2015)