Distributed Discrepancy Detection for BVR Air Combat

Justin Karneeb[1], Michael W. Floyd[1], Philip Moore[1], and David W. Aha[2]
[1]Knexus Research Corporation; Springfield, VA; USA
[2]Navy Center for Applied Research in AI; Naval Research Laboratory (Code 5514); Washington, DC; USA
{*first.last*}@knexusresearch.com | david.aha@nrl.navy.mil

Abstract

We describe an extension of the Tactical Battle Manager, which uses goal reasoning techniques to control unmanned air vehicles in simulated scenarios of beyond-visual-range air combat. Our prior work with the Tactical Battle Manager focused primarily on behavior recognition, the task of identifying the behaviors being performed by hostile aircraft. In this paper, we instead focus on distributed discrepancy detection and response. We also describe an ablation study for which we report evidence that these discrepancy management components improve mission success and efficiency.

1. Introduction

Discrepancy detection techniques can be used in goal reasoning agents to increase their ability to react to notable events and changes in their environment models (Molineaux et al. 2010). In this paper, we describe how we incorporated distributed discrepancy detection into the Tactical Battle Manager (TBM), our goal reasoning agent for controlling an Unmanned Aerial Vehicle (UAV) in simulated scenarios of beyond-visual-range (BVR) air combat (Borck et al. 2015a; Alford et al. 2015; Borck et al. 2015b).

BVR air combat is a modern style of air-to-air combat where aircraft engage each other over large distances (100km+) through the use of long-range missiles (Shaw 1985). In BVR combat, these distances afford time for reasoning because each maneuver may require a substantial duration to complete. This characteristic, together with the characteristics of these environments (e.g., imperfect information, multiagent, adversarial, continuous), make BVR air combat an interesting domain for studying goal reasoning.

In prior work (Borck et al. 2015a; Alford et al. 2015; Borck et al. 2015b), we described an earlier version of the TBM, which used behavior recognition, goal selection, and automated planning techniques to control a simulated UAV. However, goal selection was limited; it occurred only upon plan completion or when a human pilot intervened by issuing a new command to the UAV. Thus, any real-time discrepancies, such as those that may occur due to unexpected actions taken by hostile aircraft, were improperly ignored. Air combat pilots who assessed the behavior of the TBM noted that, due to this flaw, it behaved differently than human pilots.

Here we introduce a discrepancy detection capability into the TBM that takes a distributed approach in monitoring the outputs of its three main reasoning components and generating the corresponding discrepancies when appropriate. The three discrepancies are: *Model Changed* (i.e., a hostile aircraft changed its behavior), *Flanking Hostile* (i.e., the UAV is being approached by an unexpected hostile), and *Expectations Violated* (i.e., the state of the environment differs from what the UAV expects to observe). Each type of discrepancy has a unique detector that is responsible for monitoring subsystems of the TBM, identifying when discrepancies have occurred, and reevaluating the TMB's current goals and plans. This allows the TBM to dynamically respond to changing conditions in BVR combat scenarios.

We begin by describing the domain (§2) followed by a description of the TBM (§3). We then describe three new discrepancies that can be triggered and how they are processed by the TBM (§4). We next describe our empirical study on the usefulness of these discrepancies (§5), where we found evidence that incorporating their handling in the TBM improves its performance on our BVR air combat scenarios. Finally, we discuss related work (§6), plans for future work (§7), and conclude (§8).

2. Domain

As described earlier, the TBM operates in a BVR air combat domain. Compared to traditional air combat, which depends on dogfighting and rapid maneuvering to outflank the opposition, BVR combat is more deliberate. Positioning and timing is more important than low-level motion planning.

We use the Advanced Framework for Simulation, Integration and Modeling (AFSIM) system to simulate our BVR scenarios (Zeh et al. 2014). AFSIM is a high fidelity air combat simulator that allows aircraft to be controlled either programmatically via AI systems or directly by human pilots through replicated hardware (flight sticks and/or cockpit replications). Scenarios can be programmatically modified and run in batches, which helps to facilitate large-scale experimentation. In addition to this work, AFSIM is also currently being used to test how effectively human pilots are able to cooperate with the TBM in an escort mission.

AFSIM replicates real world aircraft in its scenarios. For this work, the TBMs control a modified version of an F-16 with increased speed and turning tolerances (allowing for higher G-Forces than typically allowed for human pilots). Each aircraft has a payload of eight active radar seeking missiles with an approximate firing range of 30 nautical miles. AFSIM provides the low-level controls for the aircraft allowing the TBM to focus on higher level reasoning. It provides support for flying and maintaining both absolute (i.e., global coordinates) and relative (i.e., with respect to a target) bearings as well as waypoint-based navigation. Additionally it provides real-time Weapon Engagement Zone (WEZ) information which helps the TBM determine its weapon effectiveness with respect to a given target in the current state. Finally it simplifies combat by providing actions to acquire a weapon lock and fire a weapon.

We focus our scenarios on engagements between two opposing teams of aircraft. Before each scenario, both teams are provided information on their adversaries, including their approximate whereabouts. Each team includes an airborne early warning and control (AWACS) aircraft which provides radar coverage over the entire engagement zone. In the next section we discuss the various components of the TBM and how they interact with the simulation.

3. Tactical Battle Manager (TBM)

We present an enhancement of the TBM (Figure 1), which we are developing for collaborative pilot-UAV interactions and autonomous UAV control. In the experiments described in §5, all aircraft run a version of the TBM. The TBM is split into several main categories. The Situation Assessor includes a (hostile aircraft) Behavior Recognizer and a World Modeler. The Goal Manager continually monitors the TBM for triggered discrepancies, prioritizes goals, and, if needed, selects a new goal the agent should pursue. The Predictive Planner includes, among others, a Plan Execution Predictor (PEPR) (Jensen et al. 2014) that uses a separate internal instance of AFSIM to predict the near-future plans of a hostile aircraft, and a Planner that uses this information to select plans for the UAV being controlled. Finally the Discrepancy Detectors are the new addition to the TBM that allow for a distributed means of detecting notable changes and events relevant to the system.

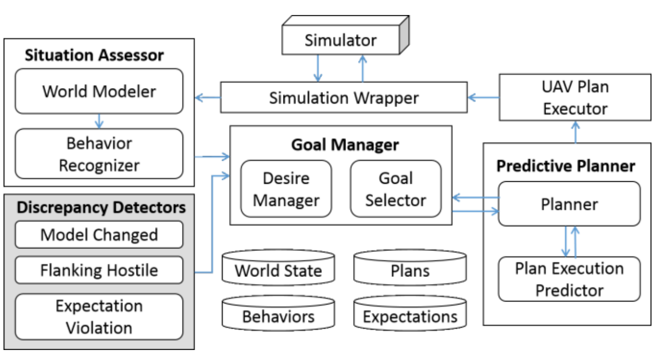

Figure 1: The Tactical Battle Manager with new discrepnacy detector components highlighted

3.1 Situation Assessor

Within the Situation Assessor, the TBM's Behavior Recognizer updates hostile aircraft models by observing their actions. It attempts to classify each hostile in terms of their aggression level and which aircraft they may be targeting. We have studied several approaches for behavior recognition, including a simple case-based approach (Borck et al. 2015a), an active behavior recognizer (Alford et al. 2015), and a policy and goal recognition algorithm (Borck et al. 2015b). However, in this paper we focus on discrepancy detection and, therefore, will use a simpler behavior recognition algorithm. In particular, we will use a rule-based algorithm to classify hostiles as *Attacking* or *Evading* a given target. This Behavior Recognizer uses a combination of a given hostile's speed, orientation, and position over time to determine if it is reacting aggressively or defensively with respect to a given ally. The TBM updates these values frequently, as they are expected to change during a mission.

3.2 Goal Manager

The Goal Manager chooses among competing goals in response to a pilot-issued command and the World State. The TBM represents a goal as a list of weighted desires and targets, where each desire's agitation is a function of the current world state. The higher a desire's agitation, the more urgently the agent wants to alleviate its symptoms (i.e., perform actions that reduce agitation). Desires range in type from *Safety*, which is agitated as the UAV moves into more dangerous situations, to *AggressivePosture*, which is agitated when the UAV is not actively engaging hostiles. Goal selection in the TBM is the process of choosing the targets, desires and desire weights that the system should focus on. This process was hand tuned over many iterations with subject domain experts on BVR combat to approximate how human pilots would prioritize their own desires. Discrepancies introduce a new means of triggering goal management that differs from our prior work, where goal selection occurred only when a goal was completed, when a goal failed, or when a pilot intervened.

3.3 Predictive Planner

The TBM's Planner is an extension of a simple plan-library planner (Borrajo et al. 2015): it selects a template, generates multiple instantiations of it, and selects the most promising instantiation. It first chooses from a set of generic symbolic plans (i.e., action sequences) that represent good BVR air combat tactics in an ungrounded state. For example, a generic plan might have the following form:

1. Approach Target Engagement
2. Close on Target to Max Missile Range

3. Acquire Missile Lock on Target
4. Fire on Target

These generic actions are then resolved into many possible grounded actions, including data such as angle of attack and engagement speed, and are given (along with the World State and Behaviors of known hostiles) to PEPR to generate a set of grounded plans with Expectations (stored as a set of expected states). Finally, the Expectations are run through the current goal's desires to determine which plan is preferable.

Finally the Discrepancy Detectors represent the newest addition to the TBM. It contains a list of components designed to monitor and generate discrepancies when appropriate. We discuss these discrepancy detectors in greater detail in the following section.

4. Discrepancy Detection

Our extension of the TBM includes three types of discrepancies that, when detected, allow the Goal Manager to react to situations in new and more efficient ways. We chose each discrepancy type to respond to failures in other TBM components and help it to recover gracefully. These three discrepancies (and the TBM components whose failure causes the discrepancies) are:

- *Model Changed* (Behavior Recognizer)
- *Flanking Hostile* (Goal Manager)
- *Expectations Violated* (Predictive Planner)

Discrepancy detection is handled in a distributed manner where each discrepancy has a corresponding detector capable of detecting it, as seen in Figure 1. The Goal Manager decides whether to respond to any discrepancies when they arise. Their distributed nature ensures that they remain agnostic to the specific implementations of the reasoning components of the TBM. Sections 4.1-4.3 provide more detail.

There is one additional type of discrepancy that the TBM processes, namely *Incoming Missile*. This discrepancy existed in prior versions of the TBM, and is processed by a reactive planner (not shown in Figure 1). Because an agent must react to this discrepancy immediately, it bypasses the TBM's deliberative reasoning cycle. In the experiments in §5, this discrepancy is always on for all aircraft, as disabling it would make battles uninteresting (i.e., because the UAV would rarely take evasive actions).

4.1 Model Changed

The Behavior Recognizer constantly monitors and re-evaluates each hostile aircraft, updating its perceived behavior every frame. PEPR uses these models when predicting plans. Thus, they are critical for generating accurate predictions/expectations. The Model Changed detector triggers on variations in the recognized behavior of hostile aircraft. When it notices that a given hostile's behavior has changed, it creates the Model Changed discrepancy and notifies the Goal Manager.

When this occurs, the Goal Manager first decides whether this hostile aircraft is relevant to its currently-executing plan. It does this by checking for the following changes between the hostile aircraft's prior and current behavior:

- Hostile's target changed to the UAV
- Hostile's target changed from the UAV
- Hostile's behavior changed (Attacking/Evading)

If these checks are all negative, the discrepancy is ignored. Otherwise, the Goal Manager is given the opportunity to select a new goal, taking into account this new information to better evaluate its desires.

4.2 Flanking Hostile

This discrepancy ensures that the UAV is never in the effective missile range of a hostile without having a chance to react (e.g., when it gets engaged by a new hostile). The Flanking Hostile detector uses pre-encoded knowledge about the maximum missile range of the hostiles and is triggered whenever the distance between the UAV and a hostile approaches this range. The Goal Manager checks whether the current plan is actively engaging this hostile, and, if not, selects a new goal and then replans.

4.3 Expectations Violated

Due to the nature of BVR air combat, plans tend to have long durations. Unfortunately, the longer a plan's duration, the harder it is for PEPR to generate accurate long-term predictions. The Expectations Violated detector alleviates this issue by triggering when PEPR's predictions differ from observations in a meaningful way. Rather than comparing the actual positions, orientations, and speeds of all aircraft in the prediction, the detector instead compares the predicted desire agitations versus the observed desire agitations for the current goal.

As an example, a common set of desires for a basic attack goal is {SafetySelf, AggressivePostureTarget, AvoidUnnecessaryTargets}. Combining these three desires favors plans that ensure the UAV's safety while aggressively pursuing a given target and avoiding other targets. Each desire is assigned a weight in the range [0, 1]. If after several minutes of plan execution, AvoidUnnecessaryTargets and SafetySelf are more agitated than expected, then Expectations Violated detector will raise a discrepancy. In practice, the discrepancy detector raises a discrepancy whenever the similarity in agitation between observed and expected states is less than 95%. For this type of discrepancy, the Goal Manager always selects a new goal and replans.

5. Empirical Study

In our empirical study we examine the following hypotheses:

H1: Extending discrepancy detection (as described in §4) will increase the TBM's mission performance

H2: Extending discrepancy detection will increase the TBM's mission efficiency

We define our metrics in §5.2. Briefly, *mission performance* is a function of the number of aircraft destroyed (on both teams), as well as the number of aircraft remaining at the end of a scenario. *Mission efficiency* is instead a function of the duration of scenarios in which the friendly team "won".

5.1 Scenarios and Evaluation Method

To test our hypotheses, we created a set of AFSIM scenarios where two teams of TBM-controlled aircraft (white and black) face off against each other in a 3v3 matchup, where each TBM is given a starting goal of destroying all opposing hostiles while minimizing allied casualties. We ran each scenario twice, once with discrepancy detection enabled only for the white team, and then again with it only enabled for the black team. During each run, we logged information about missiles shot, aircraft destroyed, and simulation length to help discern the validity of our hypotheses. An example of the starting configurations can be seen in Figure 2.

Properly testing the TBM requires a set of realistic air combat scenarios that differ sufficiently in their setup to cause varied mission tactics to be selected. Also, the number of aircraft in the scenario needs to be limited due to the computational complexity of the simulation. We settled on 3v3 because it is small enough for us to run reliably while still allowing for interesting combat tactics to emerge such as teaming up on an opponent or retreating behind an ally. It also makes it easier to recover from a disadvantage, unlike in a 2v2 scenario, where having lost an ally almost always leads to a loss.

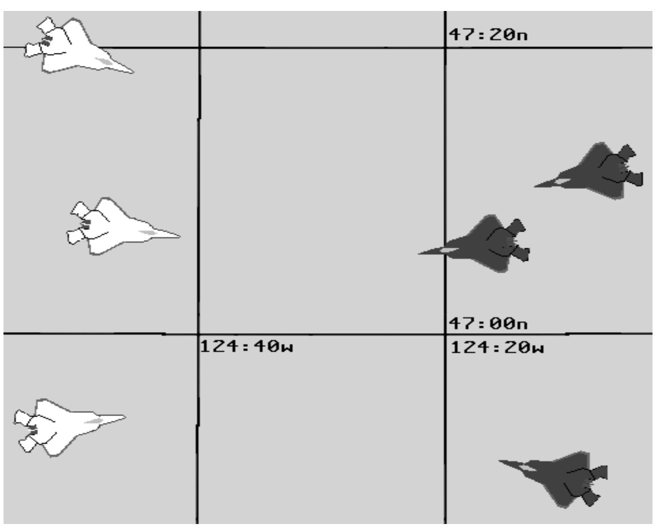

Figure 2: An example starting state for a 3v3 scenario running in the AFSIM simulator

We generated 400 unique scenarios for this study, where each is a modification of an original base scenario. In that base scenario, each team of three fighters are positioned in a line spaced 10 nautical miles apart from each other and 45 nautical miles away from the opposing team. The starting positions of each aircraft are then randomly modified by approximately 4 nautical miles in both the North/South and East/West directions. This generates varied scenarios where allies are occasionally grouped tightly and, at other times, they are spread far apart. Because discrepancy detection is enabled once for each team in each scenario, this yields a total of 800 runs. We did this to remove any inherent bias in the starting positions in the generated scenarios. The end condition for a run is either when one team is completely destroyed or 10 minutes has passed—whichever occurs first.

5.2 Metrics

To assess **H1** and **H2**, we use metrics for mission performance and mission efficiency with respect to BVR air combat.

Mission Performance
We commanded each TBM (i.e., provided initial goals) to destroy all opposing forces while minimizing ally casualties. Thus, mission performance can be measured as functions of the number of kills scored and the number of allies lost. This led to the following metrics:

- **Mission Score:** (#Hostiles Killed) - (#Allies Killed)
- **Mission Outcome:**
  - **Partial Win:** Destroyed more hostiles than allies lost
  - **Abolute Win:** Destroyed all hostiles
  - **Draw:** Equal number of hostiles and allies destroyed

If a given team reliably records a higher mission score and has more wins, we argue that it has a higher mission performance.

Mission Efficiency
Due to the nature of BVR air combat and the underlying tactics used by the TBM, it can be difficult to secure a win as described above even with superior tactics. The primary factor that determines when the TBM fires a missile is the range at which a missile is barely evadable. This mimics how a human pilot would engage, because as one combatant approaches a guaranteed kill shot, so too does an evenly matched opponent. We have found that missiles fired from the TBM have a 30% chance to destroy its target in almost all situations.

To compensate for this, we also compute mission efficiency, which we define as the speed at which a mission is won. By reacting to discrepancies, the TBM should respond to changes faster and take more opportunistic shots meaning that when it wins, it should win faster. To show this, we plan to compare the duration of Wins vs the duration of Losses, where a faster win or slower loss represents a more efficient mission.

5.3 Results

Table 1 displays the results for the 800 simulations we ran. For each metric, its corresponding tally is presented for both teams: Discrepancy Detection On (DON) and Discrepancy Detection Off (DOFF), as well as the net and percent difference of the values of each team. *In all instances,* *enabling discrepancy detection increased mission performance and mission efficiency.*

The most obvious increase is in the number of partial wins, which represent how often, at the time the mission ended (through duration or destruction) the focal team had more surviving aircraft than its opponent. Adding discrepancy detection increased the number of partial wins by 8%. It also increases the number of kills; on average, DON teams score 0.1 more kills per mission than DOFF. This seems small, but in BVR air combat any edge you can get is valuable—especially since this is facing an otherwise exact mirror (i.e., other than discrepany detection the TBMs for both teams are identical). Finally, DON teams also win about 8% more simulations via total opposition destruction. All of these differences are statistically significant as assessed via a *t*-test ($p < 0.025$ for all metrics). These results support **H1**.

Metric	DON	DOFF	Net Difference	Percent Difference
Shots	5387	5350	37	0.69%
**Kills*	1713	1634	79	4.83%
**Partial Wins*	336	311	25	8.04%
**Absolute Wins*	280	260	20	7.69%
Draws	153		---	---
** Avg. Win Duration (seconds)*	261	368	-107	-29.08%

Table 1: Results of 800 scenarios, where * denotes a statistically significant difference according to a single-tailed *t*-test ($p < 0.025$)

To assess whether Mission Efficiency is also increased (**H2**), we analyzed the average win duration metric. This value is only recorded on runs where one team achieves an absolute win (which is the only time a scenario does not reach the 10 minute time limit); it measures how efficient a team is in executing its actions. The results show that, on average, enabling discrepancy detection allows a team to win over a minute and a half faster than without it. As shown in Table 1, there is no significant difference in the total number of shots fired, nor is there a significant difference in the hit chance of a shot. Thus, we infer that a faster win time, which supports **H2**, is a result of better and more relevant plans. Even in a loss, this efficiency increase is important. Being better at evading enemies and prolonging the time before a loss could allow time for reinforcements to arrive.

5.4 Discussion

Although encouraging, there are several reasons why the increases shown are not larger. The TBM was originally designed to fly as a wingman to a human pilot in offensive air combat scenarios. As such, the logic it uses to determine when to fire a missile was designed to optimize safety above all else. Also, it currently lacks the reasoning skills to determine when it is safe to improve its shot chance by getting closer to its target. Thus, the TBM will almost always favor taking a shot from the range at which any missiles fired can be evaded if the opponent evades correctly. This ensures that, if the hostile returns fire, it should still be able to evade. Unfortuantely, this also means that when facing itself, the hostile is just as likely to evade. As shown in Table 1, the teams have similar shot-to-kill ratios. Even if discrepancies allow the TBM to take a more opportunistic shot, it still has only about a 30% chance of destroying its opponent. Additionally, a shot taken without discrepancy detection has the same odds of success. Thus, it is relatively easy for a given scenario to go awry before interesting situations occur where discrepancy detection is relevant. We plan to address this issue in future work.

6. Related Work

The focus of this research on goal reasoning agents is discrepancy detection, which has been an active research topic for several years. For example, Muñoz-Avila et al. (2010) describe the application of a goal reasoning agent, GDA-HTNBots, in a team game domain. Their agent's planner (1) generates state expectations and (2) continually monitors the state to determine if these expectations are violated (i.e., a discrepancy). When this occurs, GDA-HTNBots selects a new goal and replans accordingly. Molineaux et al.'s (2010) ARTUE agent uses a similar monitoring process for discrepancy detection. Once one is found, ARTUE abduces an explanation for it, adds any new assumptions to its beliefs, and applies a rule-based process for goal selection that examines the modified beliefs. MIDCA (Cox et al. 2012) uses A-distance to detect changes in the distribution of its beliefs over time (i.e., a discrepancy), and then assesses its cause. It uses this process to identify a new goal to select. Wilson et al.'s (2014) agent instead bounds expectations to filter minor variations from plan expectations that are not semantically meaningful; this is important for its application in a continuous domain. As a final example, GDA-C (Jaidee et al. 2013) defines a discrepancy to exist when state utility decreases, then uses a domain-specific method for calculating the discrepancy, and finally a case-based reinforcement learning algorithm to select a new goal and its associated policy. A common theme among all these agents is that they use a single method to detect discrepancies. In contrast, our extension of the TBM disburses the discrepancy detection task, where each detector monitors the output of a different component (i.e., the Behavior Recognizer, Goal Manager and Predictive Planner). This allows different types of discrepancies to be detected, which supports a more comprehensive goal reasoning process. (This may also enhance the TBM's reasoning transparency.) Also, one of the TBM's method for detecting expectation violations is unusual in that it examines differences in desire agitations rather than belief states (or their utilities).

One component of the TBM for which discrepancies are detected is its Behavior Recognizer, which models the behavior of hostile aircraft. Knowing the model of an opponent allows an agent to properly respond to it (Bowling et al. 2015), and plays a pivotal role in the TBM's Predictive Planner. We have examined multiple methods for behavior recognition in the TBM previously including PaGR (Borck et al. 2015b), which is a Policy and Goal Recognizer that can accurately recognize the complex goals and behaviors of hostile aircraft. PaGR was designed to recognize advanced tactics. However, for this paper we used a simpler rule-based behavior recognizer as it was not our current research focus, and it simplified our empirical study.

Generating successful plans for scenarios in an adversarial environment requires predicting the plans of opponents. This has been demonstrated in domains such as poker and simpler games like rock-paper-scissors (Billings et al. 1998; Tesauro 2003). This is also true for BVR air combat; TBM's Predictive Planner depends on quality opponent models to generate its predictions. In prior work, we reported that predictive planning can aid behavior recognition (Alford et al. 2015). In this paper, we instead modified the TBM's Predictive Planner to use a library of generic ungrounded plans, the recognized behaviors of its opponents, and the current goal to generate air combat plans with expected predicted states.

7. Future Work

Our future work objectives include modifications to the TBM that further increase its mission performance and efficiency. For example, we will add methods for plan generation and selection that better exploit the discrepancy detection methods described in this paper. We will also re-introduce our more complex behavior recognition algorithms, which should assist with detecting more varied and interesting discrepancies. Finally, we will extend the types of discrepancies that are generated, and develop a machine learning approach that dynamically learns new types of discrepancies and how to detect them. We have already begun to explore new discrepancies such as 'Flight Model Incorrect' (i.e., identifying when the expected capabilities of an aircraft are different than our model) and 'Weapon Engagement Zone Incorrect' (i.e., identifying when our ideal weapon range calculation is not matching observed missile performance).

In general, as the TBM adds more knowledge to its reasoning cycle, we expect new discrepancies will continue to arise. One example of this is multi-agent planning for cooperative tactics, another major area of future work for the TBM. Adding discrepancies to help identify when the current multiagent tactic is no longer relevant or to identify an opportunistic tactic is an area we are interested in exploring. One issue that remains to be solved is the increase in computational and reasoning complexity that arises from having so many agents in a single scenario. Additional work will need to be done in order to ensure that larger scale battles are feasible.

8. Conclusions

In this paper we presented methods for detecting three types of discrepancies (i.e., Model Changed, Flanking Hostile, and Expectations Violated) with a distributed approach in the Tactical Battle Manager (TBM), a goal reasoning agent. The TBM uses behavior recognition, goal management, and predictive planning techniques to control an unmanned air vehicle in beyond-visual-range air combat scenarios (implemented in a state-of-the-art simulator). We designed our discrepancy detection methods to aid in the TMB's ability to recover from an error or change in the data of its existing components. Additionally their distributed nature grants independence from the specific implementations of the TBM's reasoners, allowing us to update them without changing our detectors. In our empirical study, we found that responding to discrepancies detected by these methods significantly increased the TBM's mission performance and efficiency. We plan to add other interesting discrepancy detection and response methods to the TBM with the goal of further improving its mission performance in more comprehensive air combat scenarios.

Acknowledgements

Thanks to OSD ASD (R&E) for supporting this research. Thanks also to our subject matter experts for their many contributions.

References

Alford, R., Borck, H., Karneeb, J., & Aha, D.W. (2015). Active behavior recognition in beyond visual range air combat. In *Proceedings of the Third Conference on Advances in Cognitive Systems*. Atlanta, GA: Cognitive Systems Foundation.

Billings, D., Papp, D., Schaeffer, J., & Szafron, D. (1998). Opponent modeling in poker. *Proceedings of the Fifteenth Annual Conference on Artificial Intelligence* (pp. 493–499). Madison, WI: AAAI Press.

Borck, H., Karneeb, J., Alford, R., & Aha, D.W. (2015a). Case-based behavior recognition in beyond visual range air combat. In *Proceedings of the Twenty-Eighth Florida Artificial Intelligence Research Society Conference* (pp. 379-384). Hollywood, FL: AAAI Press.

Borck, H., Karneeb, J., Floyd, M.W., Alford, R., & Aha, D.W. (2015b). Case-based policy and goal recognition. *Proceedings of the Twenty-Third International Conference on Case-Based Reasoning* (pp. 30-43). Frankfurt, Germany: Springer.

Borrajo, D., Roubíčková, A., & Serina, I. (2015). Progress in case-based planning. *ACM Computing Surveys*, *47*(2), 35:1-35:39.

Bowling, M., Burch, N., Johanson, M., & Tammelin, O. (2015). Heads-up limit hold'em poker is solved. *Science*, *347*, 145–149.

Cox, M. T., Oates, T., Paisner, M., & Perlis, D. (2012). Noting anomalies in streams of symbolic predicates using A-distance. *Advances in Cognitive Systems*, *2*, 167-184.

Jaidee, U., Munoz-Avila, H., & Aha, D.W. (2013). Case-based goal-driven coordination of multiple learning agents. *Proceedings of the Twenty-First International Conference on Case-Based Reasoning* (pp. 164-178). Saratoga Springs, NY: Springer.

Jensen, B.A.W., Karneeb, J., Borck, H., & Aha, D.W. (2014). *Integrating AFSIM as an internal predictor* (Technical Note AIC-14-172). Washington, DC: Naval Research Laboratory, Navy Center for Applied Research in Artificial Intelligence.

Molineaux, M., Klenk, M., & Aha, D.W. (2010). Goal-driven autonomy in a Navy strategy simulation. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence* (pp. 1548-1554). Atlanta, GA: AAAI Press.

Muñoz-Avila, H., Aha, D.W., Jaidee, U., Klenk, M., & Molineaux, M. (2010). Applying goal directed autonomy to a team shooter game. *Proceedings of the Twenty-Third Florida Artificial Intelligence Research Society Conference* (pp. 465-470). Daytona Beach, FL: AAAI Press.

Shaw, R.L. (1985). Fighter combat: Tactics and maneuvering. Naval Institute Press.

Tesauro, G. (2003). Extending Q-learning to general adaptive multi-agent systems. *Advances in Neural Information Processing Systems 16* (pp. 871-878). Vancouver, British Columbia, Canada: NIPS Foundation.

Wilson, M.A., McMahon, J., & Aha, D.W. (2014). Bounded expectations for discrepancy detection in goal-driven autonomy. In A. Saffiotti, N. Hawes, G. Konidaris, & M. Tenorth (Eds.) *AI and Robotics: Papers from the AAAI Workshop* (Technical Report WS-01-14). Quebec City, Quebec, Canada: AAAI Press.

Zeh, J., Birkmire, B., Clive, P. D., Johnson, J., Krisby, A. W., Marjamaa, J. E., Miklos, L. B., Moss, M. J., & Yallaly, S. P. (2014). Advanced framework for simulation, integration, and modeling (AFSIM) version 1.8 overview (Technical Report 88ABW-2014-5041). Air Force Research Laboratory, Aerospace Systems, Wright-Patterson Air Force Base, Ohio.