Task Allocation and Motion Control for Threat-Seduction Decoys

Iman Shames, Anna Dostovalova, Jijoong Kim, Hatem Hmam

***Abstract*— Threat-seduction in its simplest form refers to the problem of luring a threat away from an asset by employing a decoy that emits a signature that is indistinguishable from the asset. In this paper, first the problem of motion control of a single decoy to achieve successful threat seduction is studied. Second, the case where multiple decoys need to respond to multiple threats is investigated and an algorithm to minimise the worst-case response time is proposed. Numerical results are presented in the end.**

I. Introduction

This paper considers the problem of protecting a non-maneaouvering surface asset from incoming threats through coordinated control of a group of decoys. The decoys are assumed to be rotary unmanned aerial vehicles that upon detection of incoming threat should position themselves in a time-optimal manner along the path of the incoming threats to be able to seduce the threat away from the asset. Such decoys are also known as break-lock decoys [1]. The problem of interest is comprised of two subproblems: (1) how one can propose a time-optimal motion strategy to position the decoy along the path of an incoming threat such that a set of operational requirements are satisfied; (2) solve an assignment problem to allocate decoys to each incoming threat such that the worst-case response time of the decoys to the threats is minimised while all the threats have decoys assigned to them. Furthermore, these two problems must be solved in real time as the engagement duration of such scenarios is often less than 120 seconds. In other words, the coordinated motion control and target assignment should be solved and executed in under 120 seconds.

There are existing technologies that address similar problems, for example one may refer to [2]–[4]. However, the biggest drawback of the existing solutions is their single-use nature and their consequent high operating costs. Thus, in this paper a solution based on the use of rotary UAVs as platforms for seduction decoys is sought. The problem studied in this paper lies in interSection of the existing literature on multi-UAV task assignments [5]–[7] and co-operative electronic warfare [8]. However, to the best of our knowledge, none of the existing results are readily capable of providing solutions that are guaranteed to satisfy the constraints necessary for successful deployment of seduction decoys in a timely fashion.

*This work is supported by Defence Science and Technology Group through Research Agreement ID6838.

Iman Shames is with Melbourne Information, Decision, and Autonomous Systems (MIDAS) Laboratory and the Department of Electrical and Electronic Engineering, University of Melbourne. Anna Dostovalova, Jijoong Kim, and Hatem Hmam are with Defence Science and Technology Group. ishames@unimelb.edu.au, {anna.dostovalova, jijoong.kim, hatem.hmam}@dst.defence.gov.au

The outline of the paper is as follows. In Section II the problem formulation and some necessary background information are presented. In Section III the minimum-time motion control problem for ensuring that a single decoy is positioned so that it can successfully seduce an incoming threat. The results presented in Section IV address the case of multiple incoming threats. The assignment problem is solved in a way that the worst-case response time of the decoys to the threats is minimised. A Numerical example is presented in Section V. Concluding remarks and future directions for research are presented in Section VI.

II. Preliminaries and Problem Formulation

Throughout this paper we assume that the dynamics of each decoy $i \in \mathcal{D}$ where $\mathcal{D} := \{1, \ldots, m\}$ is modelled as a triple integrator of the form

$$\dddot{x}^i = \gamma_i u^i, \quad x^i(0) = \bar{x}^i,\ \dot{x}^i(0) = \ddot{x}^i(0) = 0 \tag{1}$$

where $x^i \in \mathbb{R}^3$ correspond to the position i, u^i is the control input, and $\gamma_i \in \mathbb{R}^{3\times3}$ is a diagonal matrix with positive diagonal entries, and $\bar{x}^i \in \mathbb{R}^3$ are given. Furthermore, it is assumed the magnitudes of the velocity, acceleration, and the control input are bounded, i.e., $\|\dot{x}^i\|_\infty \leq \bar{x}_{1\max}$, $\|\ddot{x}^i\|_\infty \leq \bar{x}_{2\max}$, and $\|u^i\|_\infty \leq u_{\max}$ where $\bar{x}_{1\max}$, $\bar{x}_{2\max}$, and $u_{\max}$ are positive constants. The justification for this particular choice for modelling decoys lies within the fact that the proposed decoys are quadrotors whose dynamics after linearisation take the form of a triple integrator with the control input being a function of the angular velocities of its rotors, for more information the reader may refer to [9]–[12] and references therein.

Let $y^i \in \mathbb{R}^3$ denote the position of each non-manoeuvring surface asset $i \in \mathcal{A} := \{1, \ldots, p\}$ which satisfies a constant velocity trajectory, i.e.,

$$\dot{y}^i = \alpha^i, \quad y^i(0) = \bar{y}^i, \tag{2}$$

where $\alpha^i \in \mathbb{R}^3$ is constant. Note that in practice, $\begin{bmatrix}0 & 0 & 1\end{bmatrix}\alpha^i = 0$ due to the fact the asset is assumed to remain on a plane e.g. on the sea surface. The position of each threat $i \in \mathcal{T} := \{1, \ldots, n\}$ where $n \leq m$ is denoted by z^i and its motion is governed by the following set of equations

$$\dot{z}_1^i = v\cos z_4^i \cos z_5^i \tag{3a}$$
$$\dot{z}_2^i = v\sin z_4^i \cos z_5^i \tag{3b}$$
$$\dot{z}_3^i = v\sin z_5^i \tag{3c}$$
$$\dot{z}_4^i = \omega^i, \tag{3d}$$
$$\dot{z}_5^i = \eta^i, \tag{3e}$$
$$z^i(0) = \bar{z}^i, \tag{3f}$$

where $v \gg \|\alpha^i\|$, $\forall i \in \mathcal{A}$, is a known positive scalar, $\eta_i \leq \eta_{\max}$, $\omega^i \leq \omega_{\max}$, and $\bar{z}^i \in \mathbb{R}^3 \times \mathbb{S}^1 \times \mathbb{S}^1$ is the known initial position and orientation of the threat. It is assumed that the threats follow a *velocity pursuit* strategy by choosing η^i and ω^i appropriately [13][1]. Each $i \in \mathcal{T}$ drive λ_{ij} to zero where λ is the angle between $\dot{z}_{1:3}^i := [\dot{z}_1^i, \dot{z}_2^i, \dot{z}_3^i]^T$ and $\nu(y^j, z_{1:3}^i)$ where $z_{1:3}^i := [z_1^i, z_2^i, z_3^i]^T$, $j = \ell(i)$ with $\ell : \mathcal{T} \to \mathcal{A}$ being a known function that determines which threat is locked onto which asset, and

$$\nu(y^j, z_{1:3}^i) := \begin{cases} 0 & \|y^j - z_{1:3}^i\| = 0 \\ \dfrac{y^j - z_{1:3}^i}{\|y^j - z_{1:3}^i\|} & \text{otherwise.} \end{cases}$$

Proposing velocity pursuit algorithms is well-understood and is not a focus of this paper, however, we make the following definitions and assumption regarding the trajectory of the threat.

Assumption 1: Let $\zeta^i(t; \bar{z}^i, \bar{y}^j)$ denote a smooth trajectory of threat i under (2), (3), and a suitable velocity pursuit strategy where $j = \ell(i)$. The following assumptions hold.

1) There is a finite T_i, $i \in \mathcal{T}$ such that $\|\zeta_{1:3}^i(T_i) - y^j(T_i)\| \leq \bar{\epsilon}$ for some $\bar{\epsilon} > 0$.
2) The distance $\|\zeta_{1:3}^i(t; \bar{z}^i, \bar{y}^j) - y^j(t)\|$ is monotone decreasing.

Assumption 1 states that the time-of-arrival of threat i at the vicinity of asset $j = \ell(i)$, T_i, is finite. All the points in the aforementioned path belong to set $\mathcal{P}_i := \{(\bar{\zeta}_1, \bar{\zeta}_2, \bar{\zeta}_3) | \bar{\zeta} = \zeta^i(t; \bar{z}^i, \bar{y}^j), j = \ell(i), t \in [0, T_i]\}$. We present another necessary assumption next.

Assumption 2: For any $\xi \in \mathcal{P}_i$ and time $t \in [0, T_i]$ such that $\|\xi - y^j(t)\| \leq \|\zeta_{1:3}^i(t; \bar{z}^i, \bar{y}^j) - y^j(t)\|$, $\|\xi - \zeta_{1:3}^i(t; \bar{z}^i, \bar{y}^j)\|$ is monotone decreasing.

The main objective in this paper is to propose a control strategy for the decoys to place themselves along the path of the threats inside the cone in which the threats can track targets, called the *tracking cone* here, and *seduce* them away from their targets by generating a stronger radar return than that of the original targets and causing the threat to change its lock to the decoy [1]. Then, the decoy moves away from the true target. For the seduction decoy to be successful, its return signal should be stronger and indistinguishable from that of the asset from the perspective of the threat. One important parameter that ensures this to be true is the burn through range. Simply put, the burn-through range is the minimum range between a threat and an asset, for which the jamming is still effective. In what follows, we briefly introduce the concept of burn-through range. Consider the case of threat j targeting target l and seduced being by decoy i.

The burn-through range, β_{il} (in metres), is given by [1]

$$\beta_{il} = \bar{c}\sqrt{\|x_1^i - z_{1:3}^j\|} \tag{4}$$

with $\bar{c} = 10^{\frac{c+60}{40}}$ and

$$c = -71 - P_J + P_T - G_J + 2G_{TR} - G_{RJ} + 10\log A + J_R$$

where P_J is the jamming power of the decoy, P_T is the threat transmission power, G_J is the decoy antenna gain, G_{TR} is the threat's transmitter antenna gain, G_{RJ} is the jammer receiver antenna gain, J_R is the desired minimum jamming to signal ratio, and A is the area of the asset in squared meters. It is assumed that all the threats have the same power and gains and all decoys have the same power and gains. For more information refer to [1], [14].

Remark 1: The underlying assumption here is that the decoy is always transmitting with maximum power. Additionally, one can assume a variable power scenario. This however, will not fundamentally change the procedures introduced here. The reader is referred to [14] for more information.

Typically, the decoy will be placed somewhere between the threat and the asset [1], [14]. Thus, for the burn-through to occur at a distance from the target that is not larger than the distance between the target and the decoy, the following condition needs to be satisfied:

$$\beta_{il} \leq \|x_1^i - y^l\| \implies \bar{c}\sqrt{\|x_1^i - z_{1:3}^j\|} \leq \|x_1^i - y^l\|. \tag{5}$$

Another important factor for the success of a seduction decoy is that it should lie inside the tracking cone of the threat. This cone's apex is the position of the threat, its axis coincides with the velocity of the threat, and its aperture is denoted by δ and it is commonly assumed to be between 2 and 5 degrees. Fig.1 illustrates a typical scenario for the positions of the threat, the decoy, the burn-through range, and the tracking cone. The first problem to be addressed in this paper is presented below.

Problem 1: Consider the case where $\mathcal{T} = \{1, \ldots, n\}$, $\mathcal{D} = \{1, \ldots, m\}$, $\mathcal{A} = \{1, \ldots, p\}$, the decoys' dynamics are given by (1), the threats follow a velocity pursuit motion strategy, and the asset is non-manoeuvring with motion governed by (2). Let $\ell : \mathcal{T} \to \mathcal{A}$ be a known function that determines which threat is locked onto which asset. It is desired to propose a control strategy for the decoys such that there exist the set $\{\tau_i^\star\}_{i\in\mathcal{D}}$, and a one-to-one function $\sigma : \mathcal{D} \to \mathcal{T}$ such that

$$\bar{c}\sqrt{\|x^i(\tau_i^\star) - z_{1:3}^j(\tau_i^\star)\|} \leq \|x^i(\tau_i^\star) - y^l(\tau_i^\star)\| \tag{6a}$$
$$\|x^i(\tau_i^\star) - y^l(\tau_i^\star)\| \leq \|y^l(\tau_i^\star) - z_{1:3}^j(\tau_i^\star)\| \tag{6b}$$
$$\mu_{ij}(\tau_i^\star) \in [-\delta/2, \delta/2] \tag{6c}$$

[1]One may consider other guidance strategies for the threats. The choice of the strategy has no impact on the proposed solution as long as the trajectory of the threat is known.

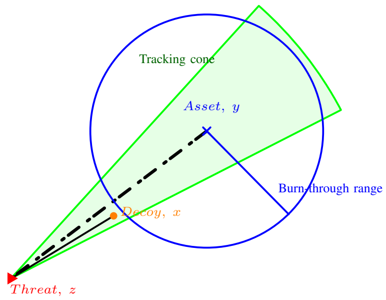

Fig. 1. An example for the positions of the threat, the decoy, the burn-through range, and the tracking cone.

where $\mu_{ij}(\tau_i^\star)$ is the angle between $\dot{z}^j(\tau_i^\star)$, $x^i(\tau_i^\star) - z_{1:3}^j(\tau_i^\star)$ in radians, $j = \sigma(i)$, and $l = \ell(j)$.

Remark 2: The inequality described (6a) corresponds to ensuring that the threat does not reach the burn-through range before passing the decoy. The inequality (6b) guarantees that the decoy is positioned between the threat and the asset. The set membership condition of (6c) is required for the threat to lock on to the decoy.

In other words, Problem 1 states a desire to devise motion strategies for the decoy such that each decoy is positioned at a location between a threat and its target in such a way that it can seduce the threat away from its target successfully.

III. The Solution for the Single-Threat Single-Decoy Scenario

In this Section we consider Problem 1 where $|\mathcal{D}| = |\mathcal{T}| = 1$ where $|\cdot|$ is the cardinality of its argument. Since, $\ell(\cdot)$ is assumed to be known, for simplicity we let $|\mathcal{A}| = 1$. For the rest of this section, we drop superscripts and subscript 1 for the index of the decoy, the threat, and the asset to simplify notation. To address Problem 1 we go one step further and find the smallest travel time for the decoy to be positioned at a point such that for some time t, (6a) – (6c) are satisfied. Call this minimum travel time τ. The problem of finding this smallest τ can be cast as the following

$$\begin{aligned} \min_{u(t),\tau} \quad & \tau \qquad\qquad (7)\\ \text{s.t.} \quad & \bar{c}\sqrt{\|x(\tau) - z_{1:3}(\tau)\|} \le \|x(\tau) - y(\tau)\| \\ & \|x(\tau) - y(\tau)\| \le \|y(\tau) - z_{1:3}(\tau)\| \\ & x(\tau) \in \mathcal{P},\ x_2(\tau) = x_3(\tau) = 0, \quad \tau \in [0, T] \\ & (1), (2), (3) \end{aligned}$$

Note that (6c) is replaced by $x(\tau) \in \mathcal{P}$ in (7). This is due to the fact that if for some time $\forall t \in [\hat{t}, T]$, $x(t) \in \mathcal{P}$ and $\|x(\hat{t}) - y(\hat{t})\| \le \|y(\hat{t}) - z_{1:3}(\hat{t})\|$ from the smoothness of $\zeta(t; \bar{z}, \bar{y})$ it follows that there exists a time $\tilde{t}$ such that $\mu(\tilde{t}) \in [-\delta/2, \delta/2]$ and $\|x(\tilde{t}) - y(\tilde{t})\| \le \|y(\hat{t}) - z_{1:3}(\hat{t})\|$. As direct consequences of Assumptions 1 and 2, we have the following remarks.

Remark 3: If for some $t \in [0, T]$ and $\xi \in \mathcal{P}$, $\|\xi - y(t)\| \le \|y(t) - z_{1:3}(t)\|$ and $\bar{c}\sqrt{\|\xi(t) - z_{1:3}(t)\|} \le \|\xi(t) - y(t)\|$ are satisfied, then, for all $\bar{t} \in [t, T]$, $\bar{c}\sqrt{\|\xi - z_{1:3}(\bar{t})\|} \le \|\xi - y(\bar{t})\|$, as long as $\|\xi - y(\bar{t})\| \le \|y(\bar{t}) - z_{1:3}(\bar{t})\|$.

Remark 4: If for some $t \in [0, T]$ and $\xi \in \mathcal{P}$, $\|\xi - y(t)\| \le \|y(t) - z_{1:3}(t)\|$ and $\bar{c}\sqrt{\|\xi(t) - z_{1:3}(t)\|} > \|\xi(t) - y(t)\|$ are satisfied, then, for all $\bar{\xi}$ such that $\|\bar{\xi} - y(t)\| \le \|\xi - y(t)\|$, $\bar{c}\sqrt{\|\xi - z_{1:3}(t)\|} > \|\xi - y(t)\|$.

Remark 3 states that if the burn-through condition (6a) is satisfied at a time t, it continues to be satisfied for any time after t. Remark 4 states that if the burn-through condition is not satisfied at a given time and location along $\mathcal{P}$, there is no other point along the path and closer to the asset for which (6a) holds.

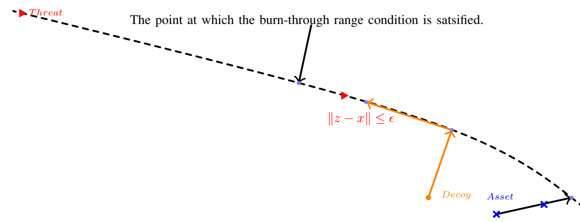

Fig. 2. First iteration of Algorithm 1 where the decoy travels to the closest point on the threat trajectory then travels along the trajectory until the algorithm returns a trajectory that satisfies the second predicate in constraint (10) of the minimum time optimal control problem (9) in Algorithm 1.

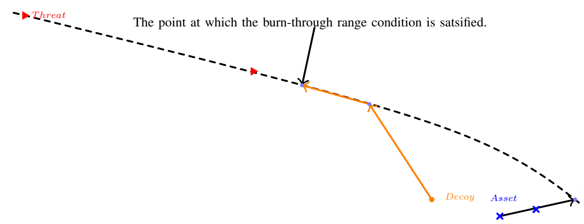

Fig. 3. Second iteration of Algorithm 1 where the decoy travels to the the last point of the decoy trajectory in the previous iteration in minimum time and then it travels along the threat path towards it. In the second iteration, the second optimisation problem in Algorithm 1 has a feasible solution due to a point that satisfies the burn-through range inequality of (6a).

Before introducing an algorithm for solving (7) we define $\theta(\xi)$ to be a unit vector in $\mathbb{R}^3$ such that $\theta(\xi)^T \dot{\zeta}_{1:3}(\kappa; \bar{z}, \bar{y}) / \|\zeta(\kappa; \bar{z}, \bar{y})\| = 1$ where $\tau \le \kappa \le T$ such that $\xi = \zeta_{1:3}(\kappa; \bar{z}, \bar{y})$. In other words, $\theta(\xi)$ corresponds to the direction of the velocity of the threat when it is at ξ.

A solution to (7) is proposed in Algorithm 1. Fig. 2 – 4 depict the outcome of three iterations of Algorithm 1. The initial point ξ_0 is assumed to satisfy the following assumption.

Assumption 3: Let $\xi_0 \in \mathcal{P}$ and $\tilde{\tau}_0$ is obtained from solving (8). It is assumed that $\exists t \in [\tilde{\tau}_0, T]$ $\|\xi_0 - y(t)\| \le \|y(t) - z_{1:3}(t)\|$.

The following result establishes how Algorithm 1 solves Problem 1 for the case where $|\mathcal{D}| = |\mathcal{T}| = 1$.

Proposition 1: Consider the optimal control problem (7). Under Assumption 3 and assuming that (7) is feasible, Algorithm 1 returns a minimum of (7) in a finite number of steps $\bar{k}$, if $\mathcal{C} \cap \mathcal{P} \subseteq \mathcal{B}$ where $\mathcal{C} := \{p \in \mathbb{R}^3 | \|p - x(0)\| \le$

Algorithm 1 Minimum time single decoy motion control: a solution to (7)

$\xi_0 \in \mathcal{P}$, $\texttt{feasible\_solution} \leftarrow \texttt{false}$, $0 < \epsilon \ll 1$
$k \leftarrow 1$
while $\neg\texttt{feasible\_solution}$ **do**

$$(\tilde{u}_k(t), \tilde{\tau}_k) \leftarrow \arg\min_{u(t),\tau} \quad \tau \tag{8}$$
such that:
$$x(0) = \bar{x}_1,\ x_2(0) = x_3(0) = 0$$
$$x(\tau) = \xi,\ x_2(\tau) = x_3(\tau) = 0$$
$$t \in [0, \tau],\ (1)$$

if $\nexists t \in [\tilde{\tau}_k, T]:\ \|x(t) - y(t)\| \leq \|y(t) - z_{1:3}(t)\|$ **then**
  break ▷ Infeasible: decoy too far away
end if

$$(\hat{x}(t), \hat{u}_k(t), \hat{\tau}_k) \leftarrow \arg\min_{u(t),\tau} \quad \tau \tag{9}$$
such that:
$$x(0) = \xi,\ x_2(0) = x_3(0) = 0$$
$$x(t) \in \mathcal{P},\ t \in [0, \hat{\tau}]$$
$$\|x(\tau) - y(\tau)\| \leq \|y(\tau) - z_{1:3}(\tau)\|$$
$$\Big(\bar{c}\|x(\tau) - z_{1:3}(\tau)\|^{\frac{1}{2}} \leq \|y(\tau) - z_{1:3}(\tau)\| \vee \|x(\tau) - z_{1:3}(\tau)\| \leq \epsilon\Big) \tag{10}$$
$$\theta(x(t))^T\dot{x}(t)/\|\dot{x}(t)\| = -1, \quad (1), (2), (3)$$

$\xi_k \leftarrow \hat{x}(\hat{\tau}_k)$
if $\hat{\tau}_k \leq \hat{\epsilon} \wedge \bar{c}\|x(\hat{\tau}_k) - z_{1:3}(\hat{\tau}_k)\|^{\frac{1}{2}} \leq \|y(\hat{\tau}_k) - z_{1:3}(\hat{\tau}_k)\|$ **then**
  $\texttt{feasible\_solution} \leftarrow \texttt{true}$
else if $\hat{\tau}_k \leq \hat{\epsilon} \wedge \bar{c}\|x(\hat{\tau}_k) - z_{1:3}(\hat{\tau}_k)\|^{\frac{1}{2}} > \|y(\hat{\tau}_k) - z_{1:3}(\hat{\tau}_k)\|$ **then**
  break ▷ Infeasible: decoy too far away
end if
$k \leftarrow k + 1$
end while
if $\texttt{feasible\_solution}$ **then**
  $\bar{k} \leftarrow k$, $\tau^\star \leftarrow \tilde{\tau}_k$, $u(t) \leftarrow \tilde{u}_k(t)$, $t \in [0, \tau^\star]$
end if

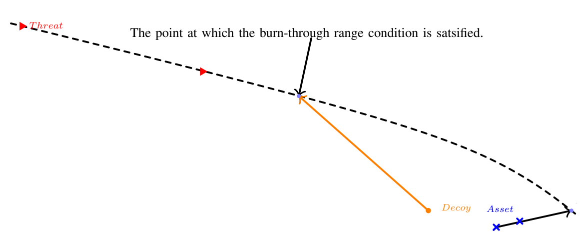

Fig. 4. Third iteration of Algorithm 1 where the decoy travels to the the last point of the decoy trajectory in the previous iteration in minimum time. The decoy does not need to move along the path towards the threat as this point satisfies (6a).

$\|\xi_{\bar{k}} - x(0)\|\}$ and $\mathcal{B} := \{p \in \mathbb{R}^3 | \|p - \zeta_{1:3}(T, \bar{z}, \bar{y})\| \leq \|\xi_{\bar{k}} - \zeta_{1:3}(T, \bar{z}, \bar{y})\|\}$.

Next, we comment on the numerical challenges of implementing Algorithm 1 and particularly on the two minimum-time optimal control problems introduced in Algorithm 1 and computing point ξ_0. Later, a modification of Algorithm 1 that dispenses with the second minimum-time optimal control of (9) along with an alternative choice of ξ_0 will be presented and it will be proven that the modified algorithm solves Problem 1 for the case where $|\mathcal{D}| = |\mathcal{T}| = 1$.

On the one hand, the minimum-time problem (8) admits a well-understood solution and is in fact a fixed-end minimum time optimal control of a chain of integrators with state constraints. On the other hand, solving (9) is not as straightforward. Nevertheless, as it will be seen shortly, for successfully solving (1), one can replace (9) with a simpler problem. That is, instead of using the control law obtained from solving (9), one can use any control law that steers the decoy from ξ along the path $\mathcal{P}$ towards the threat until some time $\hat{\tau}$ such that either $\bar{c}\|x(\hat{\tau}) - z_{1:3}(\hat{\tau})\|^{\frac{1}{2}} \leq \|y(\hat{\tau}) - z_{1:3}(\hat{\tau})\|$ or $\|x(\hat{\tau}) - z_{1:3}(\hat{\tau})\| \leq \epsilon$. In other words, (9) is replaced by the following feasibility problem.

$$\begin{aligned} \underset{\hat{u}(t),\ \hat{\tau}_k}{\text{find}} \quad & u(t) \\ \text{such that} \quad & x(0) = \xi,\ x_2(0) = x_3(0) = 0 \\ & x(t) \in \mathcal{P},\ t \in [0, \hat{\tau}_k] \\ & \|x(\hat{\tau}_k) - y(\hat{\tau}_k)\| \leq \|y(\hat{\tau}_k) - z_{1:3}(\hat{\tau}_k)\| \\ & \Big(\bar{c}\|x(\hat{\tau}_k) - z_{1:3}(\hat{\tau}_k)\|^{\frac{1}{2}} \leq \|y(\hat{\tau}_k) - z_{1:3}(\hat{\tau}_k)\| \\ & \qquad \vee \|x(\tau) - z_{1:3}(\tau)\| \leq \epsilon\Big) \\ & \theta(x(t))^T\dot{x}(t)/\|\dot{x}(t)\| = -1 \\ & (1), (2), (3) \end{aligned} \tag{11}$$

Next, we comment on computing ξ_0. This point can be computed by finding the projection of $x(0)$ onto $\mathcal{P}$. The difficulty of finding this projection is directly dependent on the geometry of $\mathcal{P}$. For example, if $\mathcal{P}$ corresponds to a line segment – often a good approximation of $\mathcal{P}$ – then the projection is straightforward. Another choice for ξ_0 is $\zeta_{1:3}(T, \bar{z}, \bar{y})$ if $\tilde{\tau}_1 < T$ where $\tilde{\tau}_1 < T$ is obtained from solving (8).

IV. Coordinated Worst-Case Positioning Time Minimisation for Multiple Threats

In this Section we consider the case where there are multiple threats and multiple decoys need to be allocated to each of the threats. An allocation algorithm that minimises the worst-case decoy positioning time is proposed. Each decoy $i \in \mathcal{D}$ solves an optimisation problem of the form (7) for each of the threats $j \in \mathcal{T}$ and obtains a corresponding $\tau_{ij}^\star$. If for some $\bar{j}$ the corresponding problem is infeasible, $\tau_{i\bar{j}}^\star$ is set to ∞. The objective of the assignment problem is to make sure that each threat is assigned at least one decoy, no decoy is assigned to more than one threat, and the assignment is carried out in a way that longest positioning time of a decoy to be able to successfully seduce its assigned threat is minimised among all possible assignments.

This assignment of the decoys to the threats can be formulated as an answer to the following optimisation problem:

$$\begin{aligned} \min \quad & \max\left(\{s_i\}_{i\in\mathcal{D}}\right) \\ \text{s.t.} \quad & s_i = \sum_{j\in\mathcal{B}} e_{ij}\tau^\star_{ij},\ i\in\mathcal{D} \\ & \sum_{j\in\mathcal{T}} e_{ij} \le 1,\ i\in\mathcal{D} \\ & \sum_{i\in\mathcal{D}} e_{ij} = 1,\ j\in\mathcal{T} \\ & e_{ij}\in\{0,1\},\ i\in\mathcal{D}, j\in\mathcal{T}. \end{aligned} \tag{12}$$

Algorithm 2 provides a solution to (12). Before continuing any further, we relate the optimisation problem (12) to finding maximal matchings with certain properties in bipartite graphs. To this aim, we first introduce the following definition.

Algorithm 2 Worst-case positioning time minimisation: a solution to (12)

Given $\mathcal{W}_0 \leftarrow \{\tau^\star_{ij}, \forall i\in\mathcal{D},\ j\in\mathcal{T}\}$
$\overline{\mathcal{D}} \leftarrow \mathcal{D},\ \overline{\mathcal{T}} \leftarrow \mathcal{T}$
$e_{ij} \leftarrow 1, \forall i\in\mathcal{D},\ j\in\mathcal{T}$
$k \leftarrow 0$
while $\left(\sum_{i\in\mathcal{D}} e_{ij} \ne 1,\ j\in\mathcal{T}\right)$ **do**
  $q \leftarrow \arg\max_{ij, i\in\mathcal{D}, j\in\mathcal{T}} \mathcal{W}_k$
  $\mathcal{W}_{k+1} \leftarrow \mathcal{W}_k \setminus \{\tau^\star_q\}$
  $e_q \leftarrow 0$
  if $\left(\exists i\in\mathcal{D} : \sum_{j\in\mathcal{T}} e_{ij} = 0\right)$ **then**
    $\overline{\mathcal{D}} \leftarrow \overline{\mathcal{D}} \setminus \{i\}$
  end if
  while $\left(\left(\exists i\in\mathcal{D} : \sum_{j\in\mathcal{T}} e_{ij} = 1\right) \wedge \left(|\overline{\mathcal{D}}| = |\overline{\mathcal{T}}|\right)\right) \vee \left(\exists j\in\mathcal{T} : \sum_{i\in\mathcal{D}} e_{ij} = 1\right)$ **do**
    if $\left(\left(\exists i\in\mathcal{D} : \sum_{j\in\mathcal{T}} e_{ij} = 1\right) \wedge \left(|\overline{\mathcal{D}}| = |\overline{\mathcal{T}}|\right)\right)$ **then**
      $\overline{\mathcal{D}} \leftarrow \overline{\mathcal{D}} \setminus \{i\}$
      $\mathcal{W}_{k+1} \leftarrow \mathcal{W}_k \setminus \{\tau^\star_{\bar{i}j}\},\ \forall \bar{i}\in\overline{\mathcal{D}}$
      $k \leftarrow k+1$
    end if
    if $\left(\exists j\in\mathcal{T} : \sum_{i\in\mathcal{D}} e_{ij} = 1\right)$ **then**
      $\overline{\mathcal{T}} \leftarrow \overline{\mathcal{T}} \setminus \{j\}$
      $\mathcal{W}_{k+1} \leftarrow \mathcal{W}_k \setminus \{\tau^\star_{i\bar{j}}\},\ \forall \bar{j}\in\overline{\mathcal{T}}$
      $k \leftarrow k+1$
    end if
    $k \leftarrow k+1$
  end while
end while

Definition 1 ([15]): Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, a matching $\mathcal{M}$ in $\mathcal{G}$ is a set of edges such that no two edges share a common vertex. A *maximal matching* is a matching $\mathcal{M}$ of a graph $\mathcal{G}$ with the property that if any edge not in $\mathcal{M}$ is added to $\mathcal{M}$, it is no longer a matching.

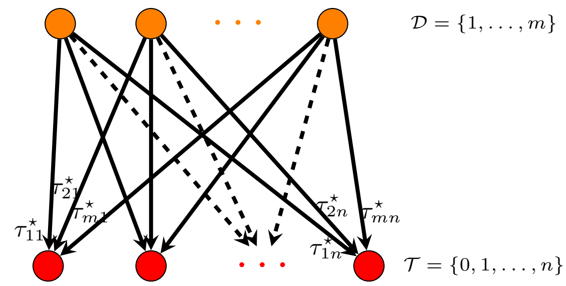

Fig. 5. Assigning decoys to threats to minimise the worst-case time.

Next, we define a weighted bi-partite graph, $\mathcal{G}_A$ with vertex set $\mathcal{V}_A = \mathcal{D}\cup\mathcal{T}$, edge set $\mathcal{E}_A = \mathcal{D}\times\mathcal{T}$, along with weights $\{\tau_{ij}\}_{i\in\mathcal{D}, j\in\mathcal{T}}$, where $|\mathcal{D}| \ge |\mathcal{T}|$. Note that (12) is equivalent to the problem of finding a maximal matching $\mathcal{M}$ such that there is no other maximal matching $\mathcal{M}'$ whose largest edge weight is strictly smaller than the largest edge weight in $\mathcal{M}$. Fig. 5 depicts the graphical representation of the assignment problem that is considered in this section.

Proposition 2: Consider the optimisation problem (12). Algorithm 2 solves this problem.
First, we show the algorithm returns a maximal matching $\mathcal{M}$ of $\mathcal{G}_A$, i.e., the edges corresponding to all $e_{ij} = 1$ form a matching. Each step of Algorithm 2 corresponds to removing one edge at a time (with a certain property as clarified below) until each vertex in $\mathcal{T}$ has an incoming edge from exactly one vertex in $\mathcal{D}$. The remaining edges clearly correspond to a maximal matching.

Next, we show that there is no other maximal matching $\mathcal{M}'$ whose largest edge weight is strictly smaller than the largest edge weight in $\mathcal{M}$. At each step the edge with the largest weight from $\mathcal{G}_A$ such that the removal of this edge does not result in impossibility of finding a matching. The edge $(\bar{i}, \bar{j})$ that leads to such impossibility for the first time belongs to the desired maximal matching. By construction all the other remaining edges have smaller weights, and there are no other matchings whose largest weight is smaller than that of $(\bar{i}, \bar{j})$. To observe the second point, assume there exists a maximal matching $\mathcal{M}'$ such that its largest weight is smaller than that of $(\bar{i}, \bar{j})$, thus $(\bar{i}, \bar{j}) \notin \mathcal{M}'$, $|\mathcal{M}'| < |\mathcal{T}|$ and consequently it is not a maximal matching. The algorithm is terminated when it is not possible to remove any edge.

Remark 5: After the first time that finding a matching becomes impossible the worst-case scenario optimisation problem is solved and the rest of the allocation can be carried out by implementing any other assignment algorithm.

V. Numerical Example

In this Section we consider a scenario where 3 threats are on track towards an asset with the speed of 274 m/s (0.8 mach) and are detected while 30 Km away from the asset. The decoys are stationed randomly at a distance of 1 Km from the target. The top speed of decoys are assumed to be 50 m/s. The objective of the scenario is that the decoys as fast as possible move to a location between the threat and the asset where the burn-through range condition (6a) is satisfied.

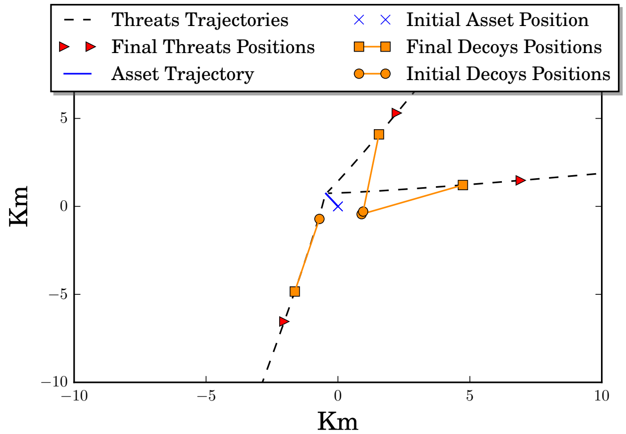

Fig. 6. Trajectories and assignments of three decoys to three incoming threats.

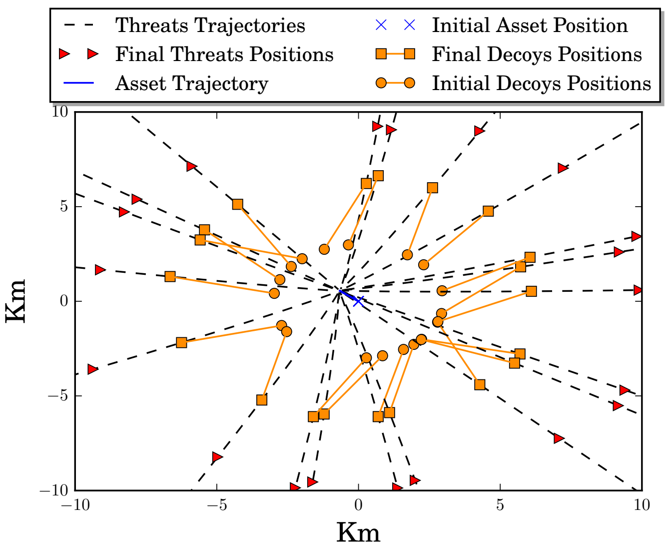

Fig. 7. Trajectories and assignments of twenty decoys to twenty incoming threats.

The arrival times of the decoys at the target are 111.40, 110.34, and 107.55 seconds. The necessary solution time for solving the assignment problem and the final destination of each decoy is 3.71 seconds and the total time for the decoys to move to their assigned destination are 84.57, 83.62, and 88.57 seconds. This will allow around 20 seconds for the seduction decoys to lure the threats away. The outcome of the scenario is depicted in Fig. 6. To demonstrate the scalability of the proposed algorithms in the next scenario we considered the case of coordination and motion control of 20 decoys in the event of 20 incoming threats. Fig. 7 depicts the outcome of this scenario.

VI. Concluding Remarks

In this paper the problem of motion control and coordination of a group of seduction decoys was considered. Particularly, a minimum-time algorithm for steering the decoys to positions along the path of a threat that enabled them to achieve their seduction objective was proposed. Furthermore, an assignment problem for minimising the worst case positioning time was introduced and solved. It was demonstrated that the assignment problem can be solved in a distributed fashion among the decoys themselves.

As an immediate future direction, the problem of coordinate pull-off problem by the group of decoys will be investigated. The objective of the problem is to ensure that all decoys project a consistent fake target for the incoming threat. Another possible future direction is providing an answer to the following problem: what is the best geometry for initial placement of the decoys around the asset?

Acknowledgements

The authors thank Dr. Farhad Farokhi for his constructive comments.

References

[1] D. L. Adamy, *EW 101, a First Course in Electronic Warfare*. Artech House, 2000.

[2] Terma skws decoy launching system. [Online]. Available: https://www.terma.com/media/118849/skws_022007.pdf

[3] S. Kerchey. Advancing soft-kill capability through improved payload positioning positioning. [Online]. Available: http://www.chemringcm.com/ /media/Files/C/Chemring-CounterMeasures-V2/press-release/mast.pdf

[4] 118 mm decoy rockets irras/dueras. Rheinmetall. [Online]. Available: http://www.angelopodesta.com/documenti/IRRAS_DUERAS.pdf

[5] E. Edison and T. Shima, "Integrated task assignment and path optimization for cooperating uninhabited aerial vehicles using genetic algorithms," *Computers & Operations Research*, vol. 38, no. 1, pp. 340–356, 2011.

[6] L. Bertuccelli, H.-L. Choi, P. Cho, and J. How, "Real-time multi-uav task assignment in dynamic and uncertain environments," in *AIAA guidance, navigation, and control conference*, 2009, p. 5776.

[7] B. Bethke, M. Valenti, and J. P. How, "Uav task assignment," *IEEE Robotics & Automation Magazine*, vol. 15, no. 1, 2008.

[8] M. J. Mears, "Cooperative electronic attack using unmanned air vehicles," in *American Control Conference, 2005. Proceedings of the 2005*. IEEE, 2005, pp. 3339–3347.

[9] R. C. Shekhar, M. Kearney, and I. Shames, "Robust model predictive control of unmanned aerial vehicles using waysets," *Journal of Guidance, Control, and Dynamics*, vol. 38, no. 10, pp. 1898–1907, 2015.

[10] C. Luis and J. L. Ny, "Design of a trajectory tracking controller for a nanoquadcopter," *arXiv preprint arXiv:1608.05786*, 2016.

[11] Z. Ju, I. Shames, and D. Nešić, "Ensuring communication connectivity in multi-agent systems in the presence of uncooperative clients," in *2016 IEEE 55th Conference on Decision and Control (CDC)*, Dec 2016, pp. 6877–6882.

[12] M. Bangura and R. Mahony, "Real-time model predictive control for quadrotors," *IFAC Proceedings Volumes*, vol. 47, no. 3, pp. 11 773–11 780, 2014.

[13] N. A. Shneydor, *Missile guidance and pursuit: kinematics, dynamics and control*. Elsevier, 1998.

[14] A. Kaptan, "Net-centric controlled distributed stand-in-jamming using uavstransmission losses and range limitations due to geo-localization problem over turkish geography," Ph.D. dissertation, Monterey, California. Naval Postgraduate School, 2012.

[15] C. Godsil and G. F. Royle, *Algebraic graph theory*. Springer Science & Business Media, 2013, vol. 207.