www.s2is.org
International Journal on Smart Sensing and Intelligent Systems
ISSN 1178-5608

AUTONOMOUS MULTI-TARGET INTERCEPTION IN DYNAMIC SETTINGS – ON-LINE PURSUER TASK ALLOCATION

Patricia Kristine Sheridan, Pawel Kosicki, Goldie Nejat*, and Beno Benhabib

Department of Mechanical and Industrial Engineering, University of Toronto, 5 King's College Road, Toronto, ON M5S 3G8, Canada

*Corresponding Author Email: nejat@mie.utoronto.ca

Submitted: Aug. 01, 2013 ***Accepted: Aug. 18, 2013*** ***Published: Sep. 05, 2013***

Abstract— In this paper, we present a generic task-allocation methodology for time-optimal, autonomous on-line interception of multiple dynamic targets by a team of robotic pursuers. The proposed novel methodology is applicable to problems consisting of numerous variations of multiple pursuers and targets. The targets are assumed to be highly maneuverable with a priori unknown, though real-time trackable, motion trajectories. Guidance theory is employed to allow each of the pursuers to navigate autonomously towards its allocated target. Numerous simulations and experiments have verified that the proposed methodology is tangibly efficient in dynamic (one-to-one) re-pairing of pursuers to targets for minimum total overall interception time.

Index terms — on-line task allocation, autonomous agents, distributed systems, robot motion planning.

I. INTRODUCTION

Teams of autonomous agents are used in applications where the task at hand may be too complex for one agent to achieve efficiently, including multi-target search and rescue, surveillance and tracking, and target interception (e.g., [1-3]). The agents (which we define here as pursuers) operate in dynamic environments, where they localize or intercept moving targets of interest. The movements of the targets are assumed to be unknown *a priori*, but, observable and trackable in real-time. In such scenarios, two problems must be addressed simultaneously: (*i*) optimal on-line (one-to-one) pairing of targets and pursuers throughout pursuit, and (*ii*) autonomous pursuer-motion planning.

In the literature, the first problem of pairing pursuers with targets has been commonly referred to as task allocation (e.g., [4-6]). Global optimality of task allocation or re-allocation (i.e., re-pairing), however, may not always be achievable as a result of the dynamic nature of the interception problem. Thus, the computational complexity of the problem would necessitate the acceptance of near-optimal pairings, as achieved subject to real-time constraints.

The second problem, pursuer motion-planning, would require a solution methodology that can easily adapt to target maneuvers as well as potential, frequent, changes in pursuer/target pairings. As shown by Kunwar [7], due to its minimal computational burden, the use of *Navigational Guidance Theory* in this endeavour would allow us to redirect limited computational resources toward the solution of the pairing problem. Furthermore, navigational guidance yields pursuer motion-trajectories that are time optimal for the interception of highly maneuverable targets [8].

In this paper, we present a generic task-allocation methodology that allows a team of autonomous robotic pursuers to optimally intercept multiple mobile targets in a time efficient manner within dynamic environments, first addressed by us in [3]. Furthermore, the proposed methodology re-pairs pursuers and targets as needed in an on-line manner. Prior to presenting our novel methodology, a discussion on the pertinent literature is provided.

a. Multi-Pursuer Task Allocation

Allocating pursuers to dynamic targets for interception is similar to the Multiple Traveling Salesperson Problem (MTSP) in 'moving cities' scenarios [9]. Solutions proposed for the MTSP with 'static cities' have been mostly off-line solutions that determine the optimal salesperson–city sequence before the salespeople begin traveling [10-14]. Such strategies, with a unique set of pairings of salespeople (pursuers) to static cities (targets) or to cities with *a priori* known 'trajectories', do not allow on-line modification of salespeople's assignments [15].

On-line solutions must ensure that pursuers always strive for optimal pairings, while adapting their motion to their assigned targets' movements. For example, a branch-and-bound solution presented by Earl *et al.* [16], where the optimal assignment is determined by selectively expanding and traversing a tree of all possible assignment sets using the A* search algorithm. Wu *et al.* [4], present an on-demand interception system where pursuers are assigned as each target's interception deadline approaches, by maximizing the probability of interception and minimizing the threat value of the target. In the work of Tsalatsanis *et al.* [6], tasks are assigned to a heterogeneous group of pursuers with different sensing capabilities by maximizing a sensor utilization function. Reimann *et al.* [17], present a differential game-theory-based re-allocation algorithm for the interception of evasive targets.

On-line assignment methods may further need to deal with scenarios in which the numbers of pursuers and targets vary dynamically. Mataric *et al*. [18] present a centralized auction method to allocate a fixed number of pursuers to static targets according to a nearest-neighbour policy, where targets pop-up at random locations. Ostergard *et al*. [19] present a distributed solution for a group of pursuers to address static targets that appear at pre-defined locations at random times. However, while these techniques account for a variable number of targets, they do not account for variability in the number of pursuers during pursuit.

b. Pursuer-Motion Planning

The efficiency of the aforementioned task-allocation task is closely related to the planning and execution of optimal pursuer-motion trajectories. Centralized multi-target interception methodologies in the literature include path planning through cell decomposition [20], polynomial interpolation trajectory generation [15], and the development of waypoints through Voronoi graphs [14]. However, while these approaches are well suited to applications where targets are static or move along pre-defined trajectories, they are not readily applicable to targets with a high degree of maneuverability.

Guidance theories, on the other hand, have been successfully used to navigate interceptors toward targets with highly-maneuverable motions. These techniques are well suited to dynamic multi-target interception problems, as they can adapt to the targets' movements quickly due to the minimal computational burden, allowing for effective continuous on-line re-planning of the robotic pursuers' movements [7, 8, 21, 22].

For scenarios in which the trajectories of targets are fixed, the avoidance of dynamic obstacles in the workspace has been addressed solely by re-planning of the paths of the pursuers when needed [23, 24]. However, it is possible that due to path deviations, a pursuer may become better suited to intercept a different target. As a result, the only manner in which to ensure that on-line path planning with obstacle avoidance does not affect the optimality of the pairing solution, is to have the pairings determined on-line as well.

II. PAIRING PROBLEM FORMULATION

The proposed methodology presented in this paper allows for the re-pairing of pursuers and targets in order to account for possible real-time variations of the interception problem due to the appearance/disappearance of targets and/or increase/reduction in the numbers of pursuers. Instantaneous position and velocity information of the pursuers and targets must be acquired or estimated for each re-pairing.

Interception is defined as a pursuer matching the position of its allocated target, and *interception time of a target by a pursuer* is the time a pursuer takes to travel from its current position to the estimated location where it *intercepts* its assigned target. Although pursuers execute their assigned motion trajectories (i.e., tasks) autonomously, the dynamic optimal allocation of tasks (i.e., continuous re-pairing) is centralized. Only one-to-one interceptions are considered. The objective, herein, is to minimize total interception time of all targets.

Let **X** be an ($m \times n$) pursuer/target pairing matrix:

$$x_{ij} = \begin{cases} 1 & \text{if Pursuer } i \text{ is paired with Target } j, \\ 0 & \text{otherwise;} \end{cases} \tag{1}$$

subject to two primary constraints:

(*a*) the total number of pairings is equal to the minimum number of pursuers or targets

$$\sum_{i=1}^{m} \sum_{j=1}^{n} x_{ij} = \min(m, n)\,, \quad \text{and}\ , \tag{2}$$

(*b*) each pursuer is paired with only one target, and vice versa

$$\sum_{j=1}^{n} x_{ij} = \begin{cases} 1 & \text{if Pursuer } i \text{ is paired,} \\ 0 & \text{otherwise;} \end{cases} \quad \forall\, i\ , \quad \text{and} \tag{3}$$

$$\sum_{i=1}^{m} x_{ij} = \begin{cases} 1 & \text{if Target j is paired,} \\ 0 & \text{otherwise;} \end{cases} \qquad \forall j \quad . \tag{4}$$

Above, *m* and *n* are the number of pursuers and targets that exist within the system at the time of the optimization, respectively. Namely, as these numbers change, pursuer/target pairings need to be re-evaluated, and **X** is resized accordingly.

Let **F** be an ($m \times n$) interception-time matrix:

$$f_{ij} = t_{ij}, \qquad \forall i,j \quad , \tag{5}$$

where t_{ij} is the time taken by Pursuer *i* to intercept Target *j*.

Let **G(X)** be a matrix representation, such that only pairings defined by **X**, are considered:

$$g_{ij} = x_{ij} \times f_{ij}, \qquad \forall i,j. \tag{6}$$

As a min-max optimization problem, our objective herein is to minimize the maximum interception time of the group of pursuers, while considering all the pursuer/target pairs, at any given time:

$$\min \ Z = \min \max[\ \mathbf{G}(\mathbf{X})\ ], \tag{7}$$

where *Z* is the objective function defined in matrix form.

III. SOLUTION METHODOLOGY

One of the primary novelties of our proposed methodology is its modular architecture, where the user can make application-specific implementation decisions within each module without affecting the functionality of the entire system. The proposed architecture utilizes four modules: Information-Acquisition, Data-Management, Task-Allocation, and Navigation, respectively, Figure 1.

a. Information-Acquisition Module

This module determines the states of the pursuers/targets within a designated workspace. It comprises a sensory system capable of determining the positions and velocities of all pursuers and targets. This module must update pursuer/target state information and the interception status, within the Data-Management module, every Δt_{state}.

b. Data-Management Module

This module acts as a common point for data sharing across the system. In order to ensure that the most recent information is available, a Blackboard Approach is used to store, update, and publish new target/pursuer state information [3]. Four specific data structures are updated onto and read from the Blackboard, as shown in Table 1.

The Interception Status, which allows this module to control the execution of the Navigation and Task-Allocation modules, indicates whether there are targets remaining to be intercepted. While the Interception Status is *false*, ($n > 0$) the module triggers the Navigation and Task-Allocation modules to continuously execute; when the Interception Status is *true* it alerts them to stop executing as the current interception pursuit has ended.

The Blackboard is updated every Δt_{state} by the Information-Acquisition module according to new pursuer and target state information, as well as latest pairings. The value of the user-selected Δt_{state} is ultimately bounded by the sensing hardware's refresh rates and the minimum detectable amount of change in position.

c. Task-Allocation Module

This module determines the pursuer/target interception pairings. Its two key components are: (*i*) the estimation of interception times, and (*ii*) the determination of optimal pairings, **X**. The first component which calculates the interception times for all possible pairings reads the pursuer/target state information from the Data-Management module and, based on the specific guidance method used, calculates and sends the matrix **F** to the optimization search engine to determine appropriate pairings.

The search engine can employ any integer-optimization method, but must be able to determine an optimal pairing solution, **X**, within Δt_{state}. The search for the new set of pairings is terminated when: (*i*) the algorithm finishes searching through all possible combinations, or (*ii*) Δt_{state} has been reached.

d. Navigation Module

This module navigates each pursuer towards its allocated target. The two components of this module, the Guidance Method and the Motion Controller, can be implemented as desired by the user provided they meet the specified update rates. The guidance method reads the target states from the Data-Management module and determines the required interception accelerations for the pursuers autonomously every Δt_{motion}, which is significantly less than Δt_{state}. These accelerations are, then, sent to the pursuer's motion controller for conversion into motion commands that are executable by the pursuers. The user-selected Δt_{motion} is bounded by the minimum pursuer position and velocity change detectable by the Information Acquisition module.

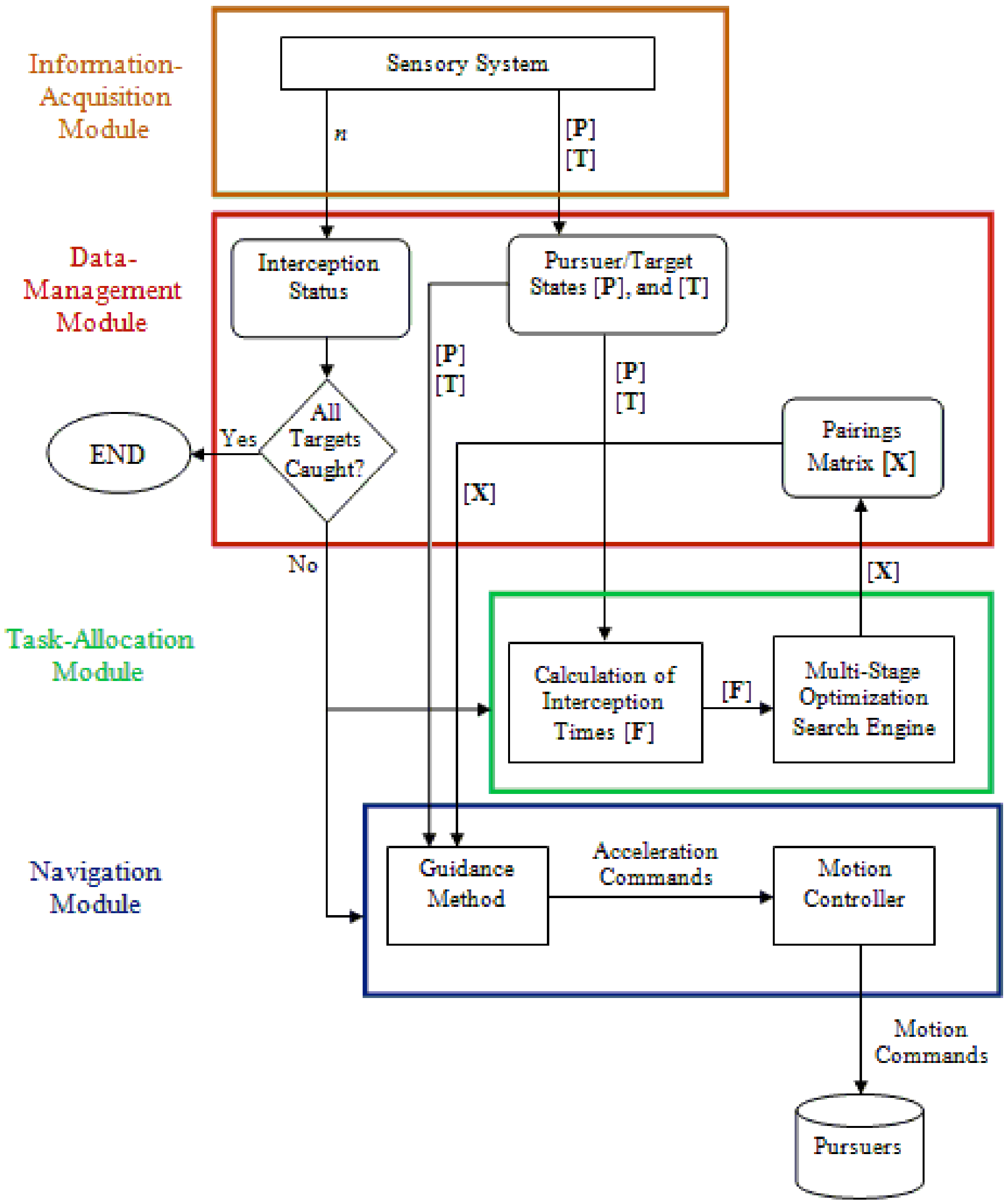


Figure 1. Architecture of Proposed Solution Methodology

Table 1: Data Exchange Rights on the Blackboard

		Data Structures*			
Module	***Purpose***	***Interception Status***	**[P]**	**[T]**	**[X]**
Information-Acquisition	Update the numbers and locations of Pursuers and Targets	W	W	W	N
Task-Allocation	Update the Pairings Matrix	R	R	R	W
Navigation	Guide Pursuers towards their Targets	R	R	R	R

*W defines write permission, R defines read permission, and N defines no permission.

IV. IMPLEMENTATION METHODOLOGY

The proposed implementation methodology addresses pursuer/target pairing, target tracking, and pursuer-motion planning.

a. Task-Allocation Module: Pursuer/target Pairing

Three main components are needed to implement this module: (*i*) a metric to evaluate the optimality of individual pairings, (*ii*) a search method to determine the optimal set of pairings, and (*iii*) a method to deal with inequalities between the number of pursuers and targets.

a.i Interception Time Metric

Interception times need to be calculated for all pursuer/target combinations in real-time due to uncertainty in target trajectories. Herein, Advanced Predictive Guidance Law (APGL) [25] is used for this purpose:

$$t_{ij} = \frac{R_{TP}}{v_R} \quad , \tag{8}$$

where v_R is defined as the magnitude of the closing velocity, $\boldsymbol{v}_c$, projected along R_{TP}, which is the Euclidean distance from the position of Pursuer *i* to that of Target *j*:

$$\boldsymbol{v}_c = \boldsymbol{v}_{T\,j} - \boldsymbol{v}_{P\,i} \quad , \tag{9}$$

$$\boldsymbol{p}_c = \boldsymbol{p}_{T\,j} - \boldsymbol{p}_{P\,i} \quad , \text{ and} \tag{10}$$

$$v_R = \frac{\boldsymbol{p}_c \cdot \boldsymbol{v}_c}{\|\boldsymbol{p}_c\|} \quad . \tag{11}$$

a.ii Optimization Search Engine

An integer-optimization method capable of providing effective results within the required update rates of the module is recommended. In our work, Simulated Annealing [26] was used.

a.iii Coping with Inequalities in the Number of Pursuers and Targets

In the case of an excess number of pursuers, allocating unpaired pursuers to either real or *fictitious* targets in the workspace could move them in an *anticipatory behaviour* of intercepting targets with inherent uncertainty in their movements. We propose the generation and use of *fictitious targets*, called *ghost targets*, that are used to anticipate where the targets will likely go, and guide pursuers towards a convergence point of all the not-intercepted targets' motions.

For Pursuer *i*, its corresponding ghost target is placed at a weighted distance, $\boldsymbol{p}_{G_i}$, amongst the remaining un-intercepted targets, as follows:

$$\boldsymbol{p}_{G_i} = \sum_{j=1}^{n} \left(\frac{\frac{1}{t_{ij}}}{\sum_{j=1}^{n} \frac{1}{t_{ij}}} \times \boldsymbol{p}_{T_j} \right) . \tag{12}$$

Adding ghost targets to the system does not affect the determination of an optimal pairing solution as the calculation of ghost targets does not occur until after a pursuer has been classified as *unpaired*.

In the case of excess targets, any target unpaired after the optimization search, is left not-pursued.

b. Information-Acquisition Module: Target Tracking

The implementation of the Information-Acquisition Module requires pursuer and target state information for frequent and consistent identification and localization, such that interception times and pursuer-motion trajectories can be estimated/planned as accurately as possible. This information can be attained via an external centralized sensory system or a distributed system of on-board-pursuer sensors depending on the application at hand.

In our experiments with mobile robots (presented in Section VI), a centralized vision system was used to acquire robot-state information. The robots were color-coded to identify their location and heading using a classification algorithm based on weighted Euclidean distances [25].

c. Navigation Module: Pursuer-Motion Planning

The implementation of the Navigation module requires a pursuer-motion planner and a motion controller. We recommend the utilization of navigation-guidance methods to be converted to appropriate motion commands in order to physically maneuver the pursuers within their environment.

c.i Guidance Method

In navigation guidance, a pursuer is accelerated toward an estimated dynamic interception point on the collision triangle between the pursuer and target [27], such that as the interception point changes, so does the acceleration applied to the pursuer. Therefore, at a required update rate, the target's velocity, acceleration, and jerk must be estimated (e.g., by using an Extended Kalman Filter with the target state information as in [7]) to determine the acceleration command for the pursuer.

The Advanced Predictive Guidance Law (APGL) was chosen in this work to better cope with highly-maneuverable targets [25]. With respect to APGL, the pursuer acceleration, $\boldsymbol{a}_P(t)$, needed to intercept a target, moving with an acceleration $\boldsymbol{a}_T$, at time instant t, is calculated by:

$$\int_0^{t_F} \boldsymbol{a}_P{}^2(t)\, dt \quad \text{, where} \tag{13}$$

$$\boldsymbol{a}_P = N\boldsymbol{v}_c\dot{\boldsymbol{\lambda}} + \frac{N}{t_{go}{}^2}\left[\frac{1-cos(\omega t_{go})}{\omega^2}\right]\boldsymbol{a}_T + \frac{N}{t_{go}{}^2}\left[\frac{\omega t_{go}-sin(\omega t_{go})}{\omega^3}\right]\dot{\boldsymbol{a}}_T \quad \text{, and} \tag{14}$$

where t_F is the total duration of the interception, N represents a navigation ratio, t_{go} is the time-to-go until interception from t to t_F, $\dot{\boldsymbol{\lambda}}$ represents the rate of change of the line-of sight, and ω is the maneuvering frequency of the target.

An obstacle-avoidance method should also be implemented to ensure that no collisions occur amongst pursuers. Any obstacle-avoidance method that minimizes deviation from the pursuer's desired direction of travel in order to avoid dynamic obstacles would be acceptable.

The Obstacle Avoidance Navigation Law (OANL) presented by Kunwar *et al.* in [25] was used in this work as it is designed to work with the APGL. The OANL uses a Collision-Cone approach to identify obstacles to be avoided. The use of obstacle avoidance does not affect the optimality of pairings as they are continuously re-evaluated on-line. Thus, if a pursuer's deviation from initial path is significant enough to result in the optimal pairing of this pursuer to an alternate target, the optimization search engine will return this alternate set of pairings.

c.ii Motion Controller

The acceleration commands that are generated by the guidance method need to be verified and translated into executable motion commands. The motion controller must ensure the acceleration commands generated are achievable by the pursuers by directly considering the kinematic model of the pursuers and the necessary non-holonomic constraints.

Feasible accelerations are converted to appropriate motion commands, and infeasible accelerations are converted to the maximum feasible accelerations which provide the least directional deviation that can be achieved while complying with the non-holonomic constraints of the pursuer.

V. SIMULATED EXPERIMENTS

Numerous simulations were carried out to examine the performance of the proposed multi-pursuer, multi-target time-optimal interception methodology. In all simulations, the initial speeds and orientations of the targets were randomly determined, and were randomly changed periodically throughout pursuit.

a. An On-line Re-pairing Example for Fixed Numbers of Pursuers and Targets

In order to determine the benefit of the proposed on-line re-pairing system (ORP), we directly compared it to both a fixed-pairing system (RFP) and a partial re-pairing system (PRP). For the RFP cases, a pursuer was allocated permanently to a target, until its interception, without the ability to be re-paired en-route, and only re-paired with a random unallocated target once its initial task was completed. For the PRP cases, whenever at least one pursuer and target were both *unallocated*, re-pairing occurred only between those unallocated pursuers and targets according to the proposed solution methodology.

In the simulations, both pursuers' and targets' starting locations were randomized, with the maximum initial distance between them limited to 3000 mm. The pursuers' and targets' maximum speed were restricted to 200 mm/s and 100 mm/s, respectively. For the ORP, Simulated Annealing was used to determine the pairings.

Table 2 shows simulation results for 200 trials, for various combinations of pursuer and target numbers that remained constant throughout the pursuit. The results clearly point to the tangible benefit of ORP over RFP.

Table 2: Mean Interception Times (s)

Number of Pursuers	***Number of Targets***	***RFP mean (sd*)***	***PRP mean (sd)***	***ORP mean (sd)***
4	8	**17.04** (4.72)	**11.13** (3.03)	**10.47** (2.66)
4	12	**25.33** (5.93)	**13.71** (3.78)	**13.20** (2.99)
4	16	**32.26** (6.68)	**16.56** (4.55)	**15.54** (3.44)

**sd*: standard deviation.

b. System Scalability

For scalability analysis, simulations were carried out with increasing quantities of pursuers and targets in order to determine the CPU and memory usage, as well as the processing time for a complete interception such that 100% of targets were intercepted, on a computer with a 2.8 GHz Intel Core i7 processor. As can be noted in Table 3, there was a minimal increase in the processor load as the number of pursuers and targets increased. However, the processing time to compute the pairings increased as the number of pursuers and targets increased. For all simulations, regardless of size, memory usage fluctuated between 200,000 Kb and 340,000 Kb. For reasonable numbers of pursuers and targets (40 or less) this system is able to compute pairings for, and navigate, pursuers in near real-time.

Table 3: CPU usage and processing time for increasing numbers of pursuer/target pairs for 20 trials - times shown in minutes

Number of Pursuers	***Number of Targets***	***Average CPU usage***	***Peak CPU usage (%)***	***Mean Interception time (sd)***	***Mean Processing time (sd)***
20	20	27%	27%	0:04.4 (0:00.9)	0:02.9 (0:00.2)
40	40	25%	27%	0:03.4 (0:00.7)	0:06.9 (0:00.2)
100	100	25%	27%	0:03.1 (0:00.4)	0:27.3 (0:01.5)
200	200	25%	27%	0:02.9 (0:00.5)	1:11.4 (0:08.3)
500	500	25%	32%	0:02.2 (0:00.4)	3:23.8 (0:20.4)
1000	1000	27%	35%	0:01.4 (0:00.2)	7:21.6 (0:37.8)

c. A Re-pairing Example for Variable Numbers of Pursuers and Targets

A detailed OPR and PRP re-pairing example is discussed herein to further demonstrate the performance of the proposed methodology. The example starts with five pursuers and five targets, whose motion characteristics are given in Table 4. In order to simulate the physical limitations of the pursuers used in experimentation (Section VI), pursuer maximum speed and lateral acceleration were restricted to 20 mm/s and 20 mm/s^2, respectively. The simulation graphs shown in Figure 2 for ORP are segmented to demonstrate the changes in pursuers' pairings and trajectories from the onset of the pursuit up until each of the defining moments of the pursuit, where an event forcing a change in pairings occurs. For the proposed solution methodology example, pairings were continually updated at least five times per second. The PRP method, shown in Figure 3, operated as described in Section V.a.

For all the figures, pursuer/target starting positions are indicated by hollow circles, the interception of a target by a pursuer is shown by a hollow square, a pursuer/target appearing during the pursuit by a solid circle, a pursuer/target disappearing during the pursuit by a solid triangle, and the instantaneous location of a ghost target by a '+' in Figure 2 only.

As can be noted through the pursuers' trajectories in Figure 2, the proposed on-line ORP methodology is quite efficient in re-pairing, especially, as pursuer/target numbers vary and the targets freely maneuver. The PRP method took 12.7 seconds to intercept all targets, while ORP required 8.2 seconds. A comparison of pairing assignments are given in Table 5.

Table 4: Target-Motion Information

Target Number	Path Followed	Max Speed (mm/s)
1	Random	10
2	Random	14
3	Random	13
4	Sinusoidal	14
5	Sinusoidal	16
6	Circular	14

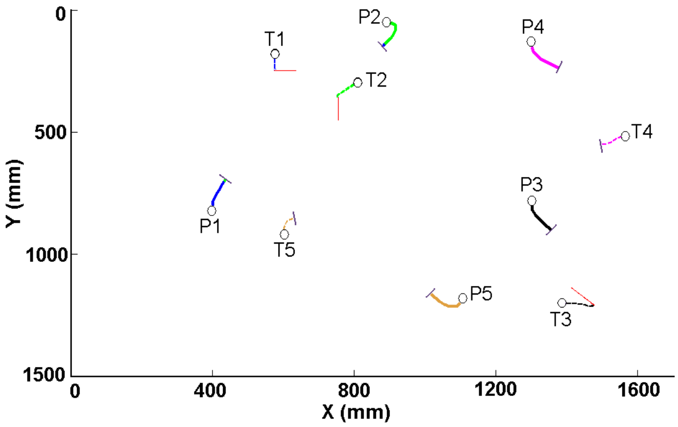


Figure 2 (a). Initial pairings: P1/T1, P2/T2, P3/T3, P4/T4, and P5/T5. At Time A, T1 turns 90° CCW, T2 turns 45° CCW, and T3 turns 120° CCW, demonstrated by the thin red lines, which cause a change in pairings to P1/T2 and P2/T1

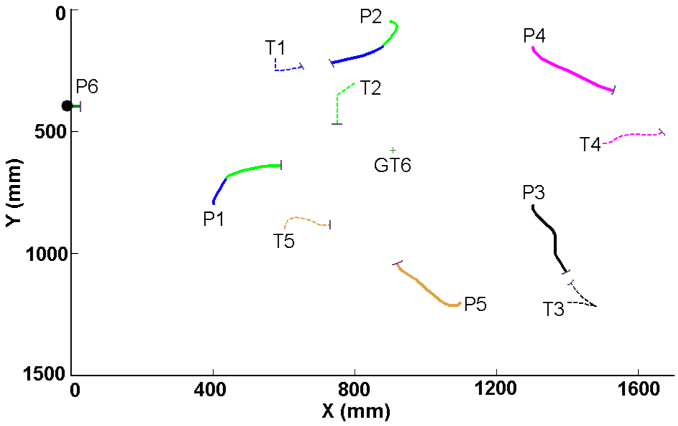


Figure 2 (b). At Time B, P6 enters the workspace. P6 is allocated to a Ghost Target GT6

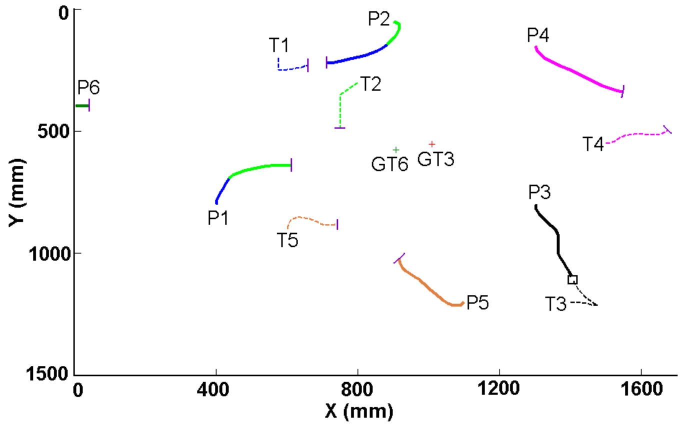


Figure 2 (c). At Time C, P3 intercepts T3 and it is re-allocated to GT3

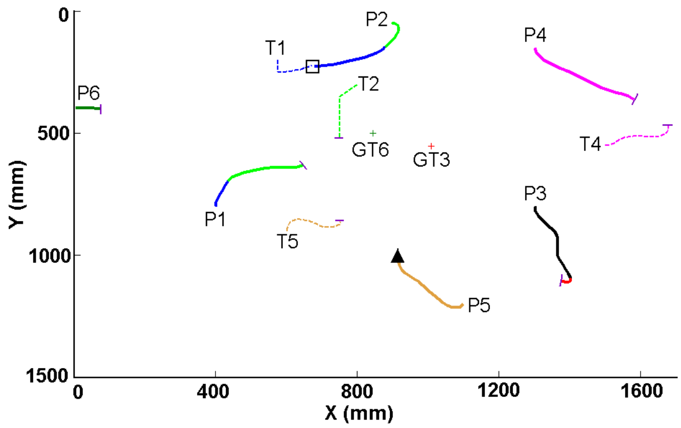


Figure 2 (d). At Time D, P2 intercepts T1, and P5 disappears. P2 is re-allocated to T2 and P1 is re-allocated to T5

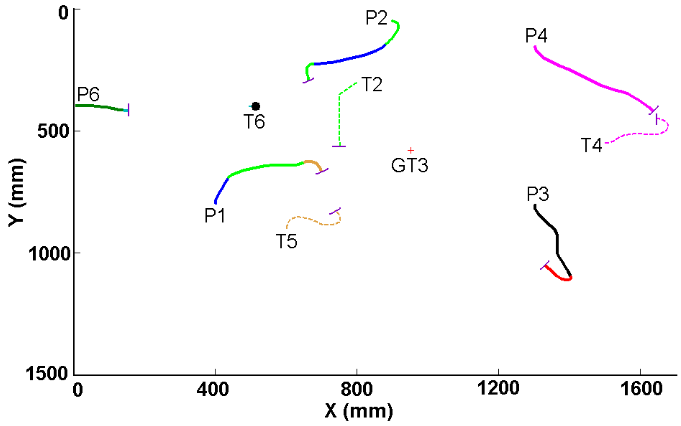


Figure 2 (e). At Time E, T6 appears. P6 is re-paired with T6

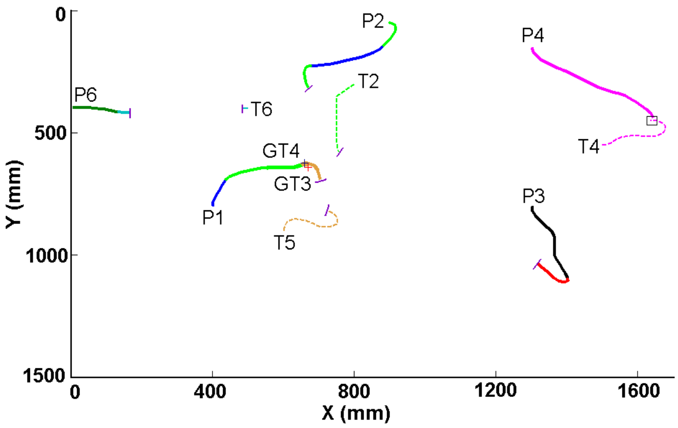


Figure 2 (f). At Time F, P4 intercepts T4. P4 is then re-allocated to GT4

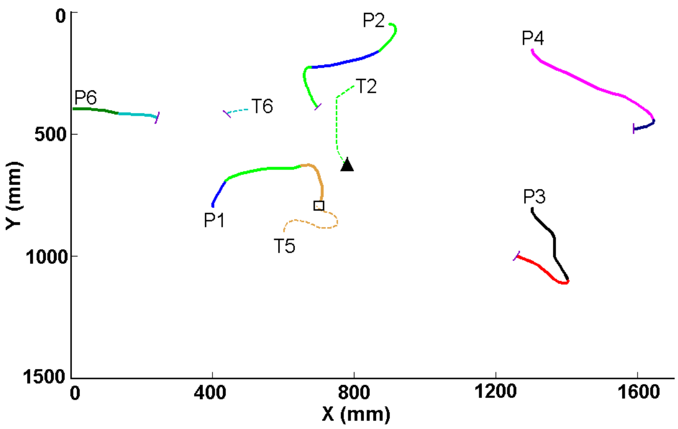


Figure 2 (g). At Time G, P1 intercepts T5, and T2 disappears. All ghost target locations collapse onto the location of T6, and it is pursued by all pursuers

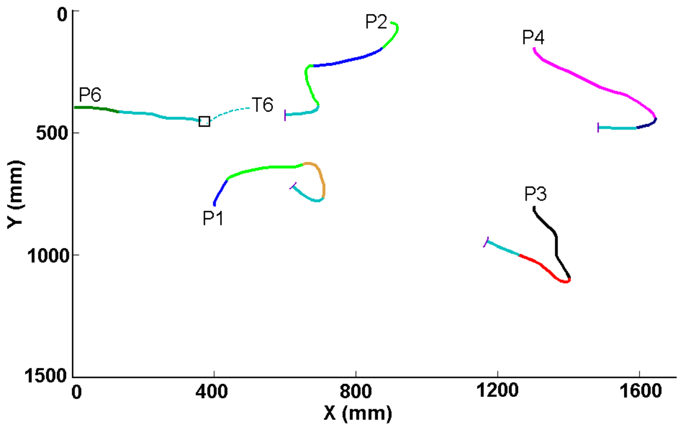


Figure 2 (h). The simulation ends when P6 intercepts T6 at Time H

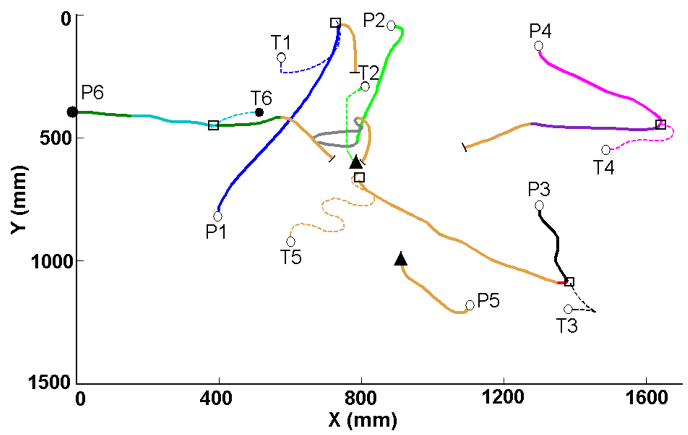


Figure 3. Simulation example using the PRP method of re-pairing

Table 5: Pursuer/target Pairing Changes for Figures 2 and 3

Time	**Changes that occur in one method**	**Pairing Changes**	
		PRP	**ORP**
A			P1/T2 P2/T1
B		P6/GT6	P6/GT6
C		P3/GT3	P3/GT3
D	ORP – P2 intercepts T1	P3/T5	P1/T5 P2/T2
E		P6/T6	P6/T6
F		P4/GT4	P4/GT4
G	ORP – P1 intercepts T5, and ghost targets collapse onto T6	P2/GT2	P1/T6 P2/T6 P3/T6 P4/T6
H		P6/GT6	End
I	PRP – P1 intercepts T1, and ghost targets collapse onto T5	P1/T5 P2/T5 P4/T5 P6/T5	
J	PRP – P3 intercepts T5	End	

VI. PHYSICAL EXPERIMENTS

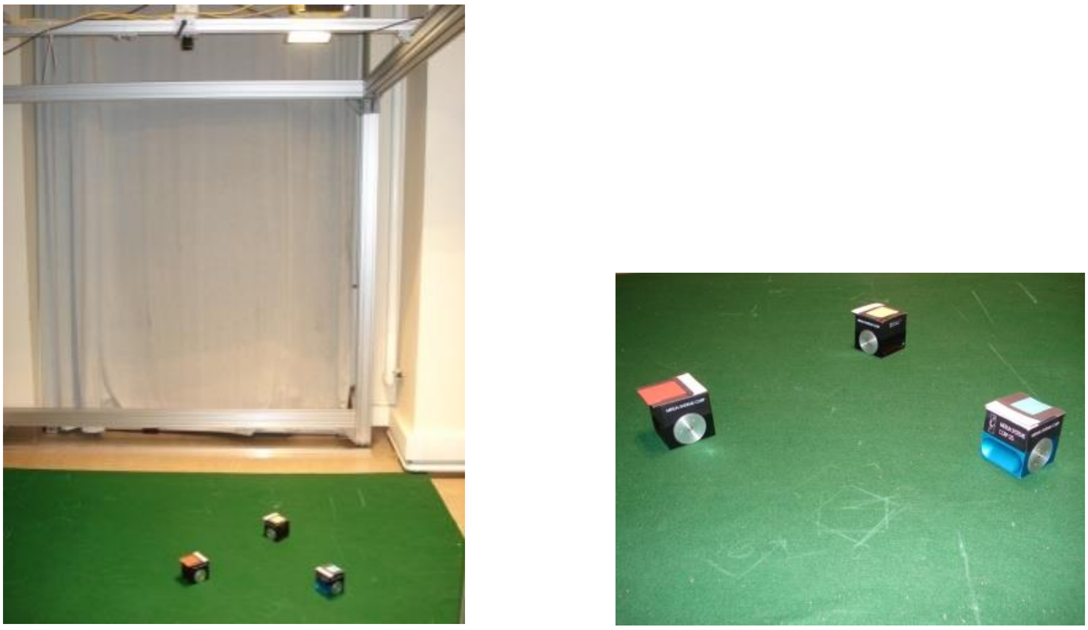
Numerous physical experiments were carried out to examine the performance of the on-line re-pairing method detailed above. Two of these experiments are presented herein.

a. Set-up and Procedure

The physical layout of the set-up is shown in Figure 4, and the hardware used is detailed in Table 6. The software is run on the same PC used in the simulations and comprises of three primary modules: (*i*) image-processing module, which extracts the colour-coded positions of the pursuer robots, (*ii*) optimal pursuer/target pairing module, and (*iii*) pursuer-motion planning and execution module, which calculates acceleration commands for all pursuers using APGL and broadcasts via a *Bluetooth* module.

Due to the limited number of (three) mobile robots, used as pursuers, the targets were 'simulated'. Namely, the (physical) robots pursued 'virtual' targets, operating in a mixed real/virtual environment. One must note that since target-state data is obtained from the Blackboard, whether it is real or simulated would not be distinguishable by the pairing algorithm nor would it affect the motion of the pursuers.

The overhead CCD camera captures images at a rate of 30 fps, and transfers them to a PC-based frame grabber. All pursuer robots implement obstacle avoidance, while, as in the simulations, their speed and lateral acceleration were restricted to 20 mm/s and 20 mm/s^2, respectively.



(*a*) **(*b*)**

Figure 4. (a) Set-up Layout, and (b) Close-up View of the Field with Pursuer Robots

Table 6: Experimental Set-up

Component	Features
Host computer Mobile Robots	Windows-based PC, with frame grabber, and Bluetooth Miabot-Pro - Bluetooth
CCD Camera	640 × 480 pixels; focal length: 6 mm; height: ~2000 mm
Floor	2000 mm × 1500 mm; Felt

For robustness analysis, the experiments were repeated three times under identical conditions (including target motion). All experiments were also simulated. The pairing and re-pairing results of the simulations and experiments were in complete agreement.

For the simulation graphs, all in-figure notations follow the same legend as that of Figure 2 in Section V.c. On the experimental results only pursuers/targets entering or exiting the scene are marked.

a.i Experiment 1

This experiment demonstrates the methodology's ability to address targets randomly appearing in or disappearing from the scene. Table 7 provides the motion characteristics of the simulated targets. Figures 5 and 6 present both the simulation and experimental results, respectively. The interception times for all three experimental runs are given in Table 8.

Table 7: Target Information for Experiment 1

Target Identity	Motion Type	Max Speed (mm/s)
1	Sinusoidal	13
2	Circular	11
3	Random	13
4	Straight*	13
5	Random	17
6	Circular	11

*The time to turn left of T4 is random.

At the start of pursuit, each pursuer is first paired with the target closest to it: P1/T1, P2/T2, and P3/T3, while the targets T4 and T5 are left not-pursued:

- At Time A, T1 and T2 pass each other, and it becomes more optimal for P1 to intercept T2, and P2 to intercept T1.
- At Time B, T6 enters the scene, and is left not-pursued as the previous pairings are deemed to be optimal.
- At Time C, P3 intercepts T3, and is re-paired with T4.
- At Time D, P1 intercepts T2, and is re-paired with T6.
- At Time E, T1 disappears, and P2 is re-paired with T5.
- At Time F, P3 intercepts T4 and is re-paired with GT3, located at the weighted distance between T5 and T6.

- At Time G, P1 intercepts T6, and there remains only one not-intercepted target. All ghost targets merge onto T5.
- At Time H, P2 intercepts T5.

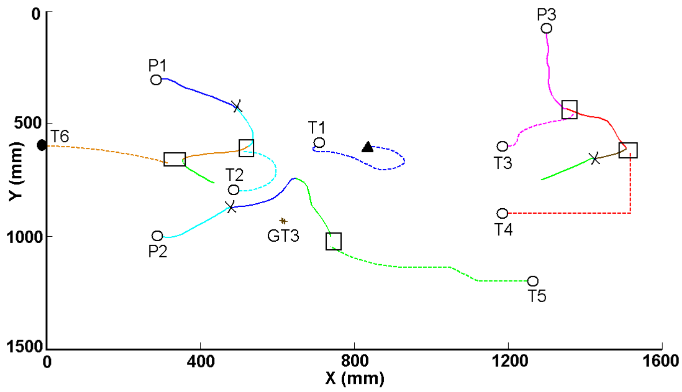


Figure 5. Pursuer Trajectories for Experiment-1 Simulation

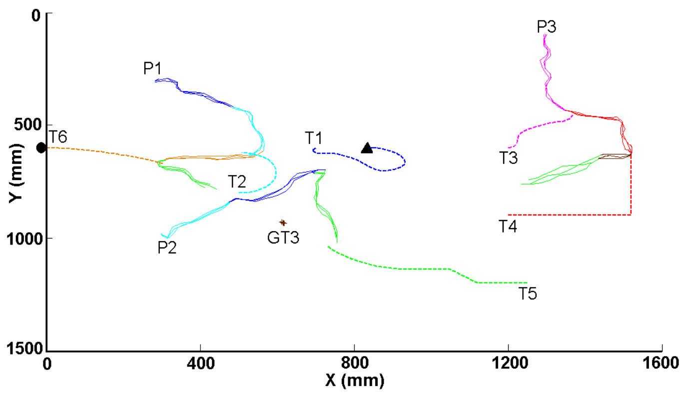


Figure 6. Pursuer Trajectories for Experiment 1, Three Experiments

Table 8: Interception Times for Experiment 1

Trial	Interception Time (s)
1	63.7
2	64.9
3	65.3

a.ii Experiment 2

Experiment 2 expands on Experiment 1 by additionally considering pursuers entering or exiting the scene. Table 9 provides the motion characteristics of the simulated targets. The simulation and experimental results are shown in Figures 7 and 8, respectively. The interception times for all three experimental runs are given in Table 10.

Table 9: Target Information for Experiment 2

Target Identity	Motion Type	Max Speed (mm/s)
1	Random	10
2	Sinusoidal	10
3	Random	8
4	Sinusoidal	12
5	Sinusoidal	10

At the start of pursuit, each pursuer is first paired with the target closest to it: P1/T1, P2/T2, while T3 and T4 are left not-pursued:

- At Time A, T5 enters the scene, and is left not-pursued as the original pairings are deemed to be optimal.
- At Time B, P2 intercepts T2, and is re-paired with T3.
- At Time C, P1 intercepts T1, and is consumed upon interception.
- At Time D, P3 enters the scene and is paired with T5.
- At Time E, P2 intercepts T3, and is re-paired with T4.
- At Time F, P3 intercepts T5, and only one target is left in the scene. Thus, T4 is pursued by all pursuers.
- At Time G, P2 intercepts T4.

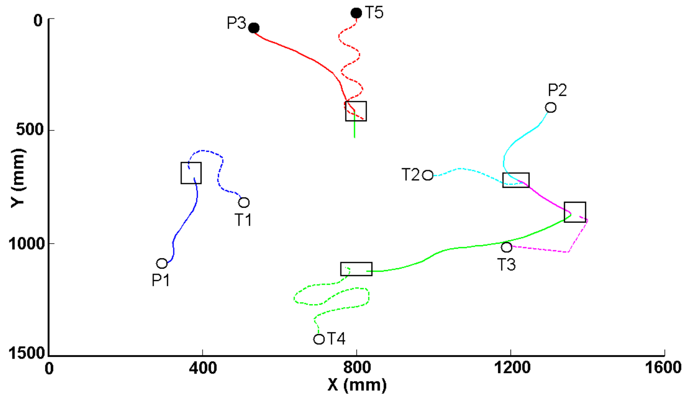


Figure 7. Pursuer Trajectories for Experiment-2 Simulation

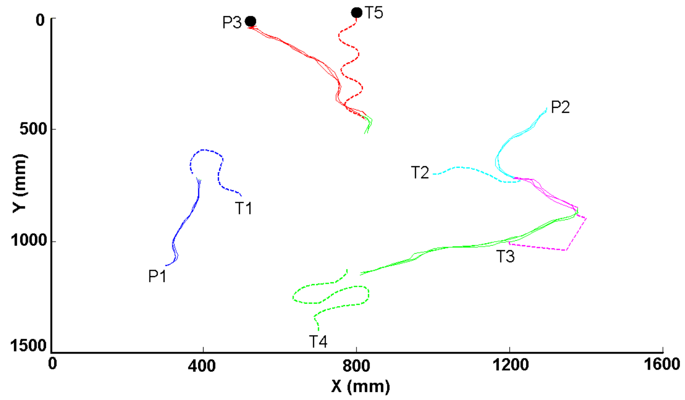


Figure 8. Pursuer Trajectories for Experiment 2, Three Experiments

For Experiments 1 and 2, during the pursuits, the pairings were re-evaluated an average of 270 and 380 times, respectively, even though a stable task-allocation is depicted.

Table 10: Interception Times for Experiment 2

Trial	Interception Time (s)
1	73.2
2	72.2
3	72.0

VII. CONCLUSIONS

A novel methodology is proposed for the on-line coordinated interception of a variable number of targets with high maneuverability by a team of pursuer robots. The solution approach comprises four modules: Task-Allocation, Navigation, Information-Acquisition, and (real-time) Data-Management via a Blackboard. Pursuer/target re-pairings are achieved by an optimization search engine in an on-line mode. Each pursuer executes its task independently and autonomously, with navigation guidance used for time-optimal motion planning.

Simulations show the proposed methodology is able to decrease interception times when compared with systems which do not constantly re-evaluate pairings for scenarios in which numbers of pursuers and targets are variable. Experiments verified the robustness of the proposed task-allocation methodology in determining optimal pairings and coordinating pursuers' motions.

Acknowledgements: *We acknowledge the assistance of Dr. Eryong She in preparation of this manuscript.*

Funding: *This work was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC).*

REFERENCES

1. Chung CF, Furukawa T. Coordinated pursuer control using particle filters for autonomous search-and-capture. Robot Auton Syst. 2009; 57:700-711.
2. Biswas S, Gupta S, Yu F, et al. Collaborative Multi-Target Tracking using Networked Micro-Robotic Vehicles. Def Transform Net-Centric Syst. 2007; 6578(12): 1-11.
3. Sheridan PK, Kosicki P, Liu C, et al. On-Line Task Allocation for the Robotic Interception of Multiple Targets in Dynamic Settings. In: IEEE/ASME Conference on Advanced Intelligent Mechatronics; 2010 July 6-9; Montreal, Canada.
4. Wu L, Xing C, Lu F, et al. An Anytime Algorithm Applied to Dynamic Weapon-Target Allocation Problem with Decreasing Weapons and Targets. In: IEEE Congress on Evolutionary Computation; 2008 June 1-6; Hong Kong.
5. Beard BW, McLain TW, Goodrich MA, et al. Coordinated Target Assignment and Intercept for Unmanned Air Vehicles. IEEE Trans. Robot Autom. 2002; 18(6): 911-

922.

6. Tsalatsanis A, Yalcin A, Valavanis KP. Optimized Task Allocation in Cooperative Robot Teams. In: Mediterranean Conference on Control & Automation; 2009 June 24-26; Thessaloniki, Greece.
7. Kunwar F, Benhabib B. Advanced Predictive Guidance Navigation for Mobile Robots: A Novel Strategy for Rendezvous in Dynamic Settings. Int. J Smart Sens Intell Syst. 2008; 1(40): 858-890.
8. Agah F, Mehrandezh M, Fenton RG, et al. On-line Robotic Interception Planning Using Rendezvous-Guidance Technique. J. Intell Robot Syst.: Theory Appl. 2004; 40(1): 23-44.
9. Helvig CS, Robins G, Zelikovsky A. Moving-Target TSP and Related Problems. In: European Symposium on Algorithms; 1998 August 24-26; Venice Italy.
10. Černý V. Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm. J Optim. Theory Appl. 1985; 45(1): 41-51.
11. Lin S, Kerninghan BW. An Effective Heuristic Algorithm for the Traveling-Salesman Problem. Oper Res. 1973; 21(2): 498-516.
12. Derr K, Manic M: Multi-Robot, Multi-Target Particle Swarm Optimization Search in Noisy Wireless Environments. In: Conference on Human System Interaction. 2009, May 21-23; Catania, Italy.
13. Chang KY, Jan GE, Su C-M, et al. Optimal Interceptions on Two-Dimensional Grids with Obstacles. J Navig. 2008; 61: 31-34.
14. McLain TW, Beard RW, Kelsey JM. Experimental Demonstration of Multiple Robot Cooperative Target Intercept. In: AIAA Guidance, Navigation, and Control Conference and Exhibit; 2002 August; Monterey, USA.
15. Flores Campos JA, Rosas Flores JA, Palacios Montufar C. Robot Trajectory Planning for Multiple 3D Moving Objects Interception: a Polynomial Interpolation Approach. In: Electronics, Robotics and Automotive Mechanics Conference; 2008 September 30 – October 4; Cuernavaca, Mexico.
16. Earl MG, D'Andrea R. A decomposition approach to multi-vehicle cooperative control. Robot Auton Syst. 2007; 55(4): 276-291.
17. Reimann J, Vachtsevanos G. UAVs in Urban Operations: Target Interception and Containment. J Intell Robot Syst. 2006; 47(4): 383–396.
18. Mataric MJ, Sukhatme GS, Ostergaard EH. Multi-Robot Task Allocation in Uncertain

Environments. Auton Robot. 2003; 14: 255-263.

19. Ostergard EH, Mataric MJ, Sukhatme GS. Distributed multi-robot task allocation for emergency handling. In: IEEE/RSJ International Conference on Intelligent Robots and Systems. 2001 October 29 – November 3; Hawaii USA.
20. Ferrari S, Cai C, Fierro R, et al. A Geometric Optimization Approach to Detecting and Intercepting. In: American Control Conference. 2007 July 11-13; New York, USA.
21. Borg JM, Mehrandezh M, Fenton RG, et al. Navigation-Guidance-Based Robotic Interception of Moving Objects in Industrial Settings. J Intell Robot Syst. 2002; 33(1): 1-23.
22. Mehrandezh M, Sela MN, Fenton RG, et al. Robotic Interception of Moving Objects Using an Augmented Ideal Proportional Navigation Guidance Technique. IEEE Trans on Syst Man Cybern. 2000; 30(3): 238-250.
23. Shetty VK, Sudit M, Nagi R. Priority-based assignment and routing of a fleet of unmanned combat aerial vehicles. Comput Oper Res. 2008; 25:1813-1828.
24. Zhu R, Sun D, Zhou Z. Cooperation Strategy of Unmanned Air Vehicles for Multitarget Interception. J Guid, Control, and Dyn. 2005; 28(5): 1068-1072.
25. Kunwar F, Sheridan PK, Benhabib B. Predictive Guidance-Based Navigation for Mobile Robots: A Novel Strategy for Target Interception on Realistic Terrains. J Intell Robot Syst. 2010; 59(3-4): 367-398.
26. Kirkpatrick S. Optimization by Simulated Annealing: Quantitative Studies. J Stat Phys. 1984; 34(5-6): 975-986.
27. Shneydor NA. Missile Guidance and Pursuit Kinematics, Dynamics, and Control. Horwood Publishing Limited: Great Britain; 1998. p.78-79.