# A New Conflict Resolution Method for Multiple-Mobile Robots in Cluttered Environments with Motion-Liveness

Mohammadali Shahriari, *Student Member, IEEE,* Mohammad Biglarbegian, *Senior Member, IEEE*

***Abstract*—This paper presents a new conflict resolution methodology for multiple-mobile robots while ensuring their motion-liveness, especially for cluttered and dynamic environments. Our method constructs a mathematical formulation in a form of an optimization problem by minimizing the overall travel times of the robots subject to resolving all the conflicts in their motion. This optimization problem can be easily solved through coordinating only the robots' speeds. To overcome the computational cost in executing the algorithm for very cluttered environments, we develop an innovative method through clustering the environment into independent sub-problems that can be solved using parallel programming techniques. We demonstrate the scalability of our approach through performing extensive simulations. Simulation results showed that our proposed method is capable of resolving the conflicts of 100 robots in less than 1.23 seconds in a cluttered environment that has 4357 intersections in the paths of the robots. We also developed an experimental testbed and demonstrated that our approach can be implemented in real-time. We finally compared our approach with other existing methods in the literature both quantitatively and qualitatively. This comparison shows while our approach is mathematically sound, it is more computationally efficient, scalable for very large number of robots, and guarantees the live and smooth motion of robots.**

***Index Terms*—Multiple Mobile Robots, Conflict Resolution, Collision Avoidance, Meta-heuristic Optimization, Motion-liveness.**

## I. Introduction

Multiple-Mobile Robots (MMRs) have vital applications in search and rescue, surveillance, cooperative robotics, and scientific data collection [1–5]. Many of these applications require collision-free movement of robots, which allows for safe and efficient operation, especially in cluttered environments [6–9].

Different well-known approaches have been used for conflict resolution of MMRs such as incorporation of virtual force [10], behavioral and avoidance potential functions [11], [12], flocking algorithms [13] and centralized and decentralized coordination [14], [15]. These approaches result in collision-free motion of robots in small scale systems; however, they have the known issue of local minima. Even though some works have addressed the local minima, no results have shown that these methods can cope with a large number of robots under continuous motion without stopping or delays, especially in cluttered environments [16–18]. Strategic decision rules in agent-based frameworks are also promising approaches with a system perspective that can be used in MMRs conflict resolution [19], but in practice the main challenge is to model MMRs as multi-agents in order to adapt the strategic decisions rules for MMRs conflict resolution. Many studies extended the aforementioned methods to address the stated problem of the local minima. We divide these approaches into two main strategies: path coordination and velocity coordination.

Path coordination for conflict resolution such as low level navigations [20] works well with a small number of robots, but fails to resolve conflicts when the environment is cluttered. Randomized motion planning techniques plan collision-free motion through constructing roadmap for robots and it is shown to be scalable [21]; however, the computation time of these methods are high. Other path coordination approach is prioritized planning [16] that uses behavioural algorithms with priority scheme to find collision-free motion. It was shown in [22] that this method can be scalable for a large number of robots, but it suffers from a known issue of deadlock, which has not been addressed properly especially when dealing with cluttered environments. A deadlock is a state where robots are forced to stop or move at a very low speed to avoid collisions. The task allocation approach [23] and cooperative navigation [24] are also shown to be scalable but have the same problem of deadlock. Velocity coordination techniques adjust the speed of the robots in order to avoid collisions. Enforcing velocity coordination in the path of the robots will provide more flexibility in motion [25]. A decoupled coordinated trajectory planning was proposed in [26] using pre-specified paths and scheduling the motions of the robots along their paths, but for practical applications the path specification procedure and deadlock problem of more than three robots were not discussed. Reciprocal algorithms such as [27] are velocity coordination techniques where each robot adjusts its own velocity by observing other robots' velocities and shown to be scalable [28]. Cooperative collision avoidance techniques based on the sensor networks are also practical solutions to facilitate collision avoidance [29], which assures safety while deadlocks are the main issues. There are also control based techniques with implicit satisfaction for safety and live motion, but to the best of our knowledge these methods cannot maintain live motion in cluttered environments [30].

Within the context of conflict resolution, one parameter that needs to be satisfied is motion-liveness. Motion-liveness ensures the continuous motion flow of MMRs without stopping. Satisfying motion-liveness is important, especially in

M. Shahriari and M. Biglarbegian are with the School of Engineering, University of Guelph, Canada, e-mail: mshahria@uoguelph.ca, and mbiglarb@uoguelph.ca.

cluttered environments with a large number of robots, but this makes conflict resolution much more challenging [31], [32]. While many studies have been done on collision avoidance to guarantee motion safety, very few of them have considered motion-liveness in their approaches [11], [31], [33]. Motion-liveness, as discussed in [34], is critical when one is dealing with MMRs in a cluttered environment because of deadlocks in conflict zones. To overcome the deadlocks in MMRs, several models have been developed including partitioning the planar space [35], treating robots as unexpected obstacles [36], and using real-time management of resource allocation systems [37]. In another work, a conflict resolution method was developed by combining priority schemes and multilevel conflict resolution [38]. The computational complexity of this problem was the main issue that needed to be addressed.

Scalability is another factor that needs to be considered in developing conflict resolution, particularly for cluttered environments. While scalability has been reported in [16], [22], [28], [33], [39], [40], only few methods have considered motion-liveness [27], [37], [41]. Amongst the methods that are scalable, [40] and [42] studied dynamic environments, cluttered environments were considered in [16], [24], [28], and motion-liveness was addressed in [31], [41], [42]. However, computational cost remains the main issue of these works, notably in cluttered environments. One approach to deal with large-scale systems is using meta-heuristic optimization techniques [43], such as genetic algorithm (GA) [44], and Particle Swarm Optimization (PSO) [45]. These works, in spite of developing a systematic way to resolve the conflicts and taking robots' motion-liveness into consideration, did not address the challenges of complexity in cluttered environments. For example, conflict resolution was modeled as a well-known scheduling problem in [33], but the combinatorial complexity of the scheduling limited the applicability of this approach. An immune algorithm [46] was shown to work better than PSO and GA; however, only a system of two robots was studied. In [47], the ant colony optimization algorithm was used for on-line collision avoidance, nevertheless, it is an open question whether this method can be implemented in real-time because of its computational cost. To the best of our knowledge, none of the works in the literature has been able to develop a method that generates collision-free motion of MMRs in cluttered environments with live and smooth motion, that is computationally efficient for real-time implementation.

In this paper, we develop a new velocity based strategy for conflict resolution of MMRs while ensuring motion-liveness, especially for cluttered environments with a large number of robots' paths intersections. We propose a strategy that is scalable for a large number of robots and can be performed in real-time and in dynamic environments. As for the contributions of this paper, (i) we propose a new methodology of conflict resolution for MMRs with smooth paths and motion-liveness in dynamic and cluttered environments, (ii) we formulate conflict resolution as an optimization problem that considers robots motion constraints, (iii) we develop a novel environment clustering approach for conflict resolution that improves computational efficiency in cluttered environments which can be implemented in real-time, (iv) we present extensive simulations using optimization techniques to demonstrate the effectiveness of our proposed strategies, and (v) we also develop an experimental testbed to validate our approach.

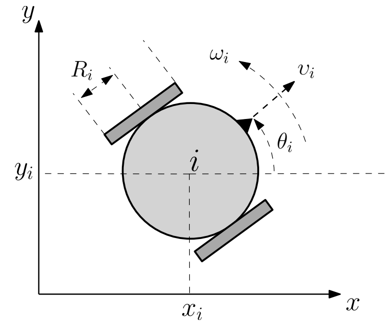



Fig. 1: Global coordinates and mobile robot parameters.

The rest of the paper is organized as follows: Section II provides the path planning strategy for MMRs, Section III presents the conflict resolution strategy, Section IV presents the simulation and experimental results, and Section V concludes the paper.

## II. Path Planning

The main contribution of this paper is developing conflict resolution strategies. However, to be able to perform these methods we need to obtain the robots paths first and then coordinate the robots trajectories. In this section, we briefly present our path planning approach for non-holonomic mobile robots.

Throughout this paper, the parameters of the $i^{th}$ robot are shown by the index $i$ ($i = 1, ..., n$), where $n$ is the number of MMRs. A non-holonomic mobile robot with its kinematic and geometrical parameters is shown in Fig.1. The coordinate frame $\{x, y\}$ is used to show the position and orientation of the $i^{th}$ robot. The equation of the $i^{th}$ robot's motion in the global coordinate frame is expressed as follows:

$$\begin{aligned} \dot{x}_i &= \upsilon_i \cos\theta_i \\ \dot{y}_i &= \upsilon_i \sin\theta_i \\ \dot{\theta}_i &= \omega_i \end{aligned} \tag{1}$$

where $x_i$, $y_i$, and $\theta_i$ are positions and orientation of the $i^{th}$ robot, respectively, $\dot{x}_i$, $\dot{y}_i$ are the translational velocities, $\dot{\theta}_i$ is the angular velocity, and $\upsilon_i$ and $\omega_i$ are translational and angular velocities, respectively. The non-holonomic constraint for the $i^{th}$ robot is expressed as

$$\dot{x}_i \sin\theta_i - \dot{y}_i \cos\theta_i = 0 \tag{2}$$

The constraint given in (2) will be used later in Section III-D to generate feasible paths for non-holonomic MMRs.

In our proposed method of path planning, we find the shortest paths for the robots satisfying their motion constraints while avoiding obstacles. Therefore, we define a path for the $i^{th}$ robot by a set of points $\boldsymbol{P}_i = \{\boldsymbol{p}_{i,0}, \boldsymbol{p}_{i,1}, ..., \boldsymbol{p}_{i,q}\}$ starting from the robot's starting point ($\boldsymbol{p}_{i,0}$) to its target ($\boldsymbol{p}_{i,q}$), as shown in Fig 2. The number of points, $q$, is chosen according to the complexity of the environment, i.e., the number and the

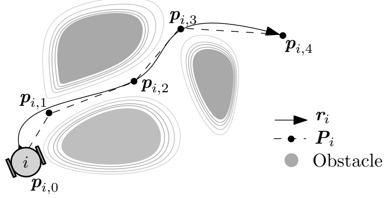



Fig. 2: Path planning and smoothing.

position of the obstacles. After finding $\boldsymbol{P}_i$ for all the robots, we smooth the paths by spline fitting. We initially start with a small number of points, e.g., $q = 4$, and then increase it to find the optimal path. For the obstacle avoidance, we calculate a matrix map of the environment. The matrix map assigns values to all the positions in the environment quantifying how close the position is to obstacles. To find the matrix map of the environment, we use the value of repulsive potential function [48] at a position $(x, y)$ of the environment given by

$$g(x, y) = \sum_{o=1}^{n_o} \frac{w}{\min ||[x, y]' - \boldsymbol{I}_o||} \tag{3}$$

where $g(x, y)$ is the repulsive potential function, $x$ and $y$ denote a position of any point in the environment, $w$ is a constant value, $o$ is the index of an obstacle, $n_o$ is the number of obstacles, $||.||$ is a Euclidean norm, and $\boldsymbol{I}_o$ is the set of border coordinates of the $o^{th}$ obstacle. We use a large number for $w$, e.g., $10^3$, and use this constant as a weighting penalty if a path is too close to an obstacle resulting in a high value for $g$. The path planning problem is now defined as minimizing the distance between the points in $\boldsymbol{P}_i$ and also minimizing the maximum value of $g(x, y)$ through the path by (3) as follows:

$$\underset{\boldsymbol{p}_{i,0}, \boldsymbol{p}_{i,1}, \ldots, \boldsymbol{p}_{i,q}}{\text{minimize}} \left( \sum_{z=1}^{q} \|\boldsymbol{p}_{i,z} - \boldsymbol{p}_{i,z-1}\| + \max_{x,y \in \boldsymbol{A}_i} g(x, y) \right) \tag{4}$$

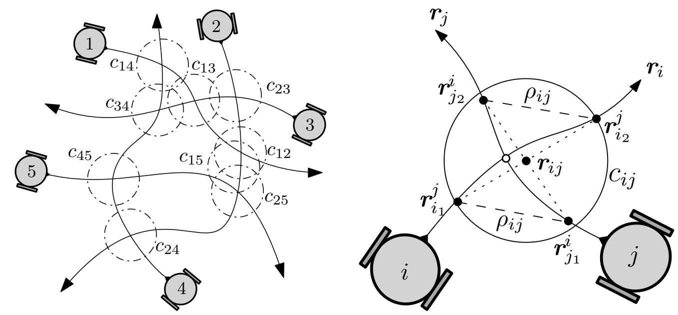
where $\boldsymbol{A}_i$ is a set of all the connecting points from $\boldsymbol{p}_{i,0}$ to $\boldsymbol{p}_{i,q}$. We use Genetic Algorithm (GA) to solve (4). $\boldsymbol{P}_i$ is initialized by a set of waypoints on a straight line connecting $\boldsymbol{p}_{i,0}$ to $\boldsymbol{p}_{i,q}$. After finding the paths points, $\boldsymbol{P}_i$, we smooth the paths as shown in Fig. 2. In this figure, the potential repulsive function, $g(x, y)$, is shown with contours around the obstacles. To smooth the paths, we consider the path of the $i^{th}$ robot, $\boldsymbol{r}_i$, to be a spline of order $m$ given by the following expression:

$$\boldsymbol{r}_i(\gamma_i) = x_i(\gamma_i)\boldsymbol{i} + y_i(\gamma_i)\boldsymbol{j} \tag{5}$$

where

$$x_i(\gamma_i) = \sum_{d=0}^{m} \alpha_{d,i}\gamma_i^d, \;\; y_i(\gamma_i) = \sum_{d=0}^{m} \beta_{d,i}\gamma_i^d, \;\; \gamma_i \in [0, 1] \tag{6}$$

where $\alpha_{0,i}, ..., \alpha_{m,i}, \beta_{0,i}, ..., \beta_{m,i}$ are spline's coefficients of order $m$. These coefficients are obtained by fitting (5) on the robots' paths points $\boldsymbol{P}_i$, subjected to the robots' velocity constraints, which are explained further in Section III-D. The order of spline $m$ is chosen based on the complexity of the robot's path, e.g., between 3 to 7. In (5), $\gamma_i$ is the parameter that we use to adjust the robots speeds in our conflict resolution methodology, which will be elaborated in Section III-A.




(a) Potential collision zones.

(b) Potential collision point and safety radius $\rho_{ij}$.

Fig. 3: Potential collision zones in a cluttered environment.

To simplify the notations, we will consider $\boldsymbol{r}_i$, $x_i$, $y_i$ as short for $\boldsymbol{r}_i(\gamma_i)$, $x_i(\gamma_i)$, $y_i(\gamma_i)$, respectively.

## III. Conflict Resolution for MMRs

This section presents our developed methodologies for conflict resolution of MMRs. These methods are based on a mathematical formulation that can also provide robots with smooth and live motion. In this paper, using our strategy shown in Section II, we assume the trajectories of the robots are known and then we develop techniques for conflict resolution by navigating the robots through only adjusting their speeds. This is the emphasis of our work.

It should be noted that only changing the robots' paths will not effectively solve the conflict resolution, especially for cluttered environments. Using a velocity approach, we can resolve the conflicts efficiently while guaranteeing motion-liveness (as will be shown later). In this paper, we find the paths for each robot, as presented in Section II, and then perform conflict resolution with velocity profiles without changing the paths. It should be noted that even if the paths of the robots are fixed, coordinating an MMR system is still very challenging because of existence of a lot of potential collision zones.

In this section, we first define the velocity rates to adjust the robots' speeds, and then formulate collision-free motion as a function of velocity rates by finding potential collision zones.

### A. Potential Collision Zones

By considering $\gamma_i$ to be a continuous function of time, we can adjust the speed of the robots in order to navigate them safely (to avoid collision) without stopping them (to satisfy motion-liveness). We assume $\gamma_i$ to be a linear function of the time as follows:[1]:

$$\gamma_i = k_i t \qquad k_i \in \left[\frac{1}{t_{u_i}}, \frac{1}{t_{l_i}}\right] \tag{7}$$

[1]We also considered $\gamma$ to be a quadratic function of time; details and derivations are presented in Appendix A. We showed that when using a quadratic form, the formulation becomes complex and computational cost will be higher compared to a linear function. Therefore, it is not suitable for cluttered fast changing environments.

where $t$ represents time, $k_i$ indicates the $i^{\text{th}}$ robot's velocity rate parameter (the $i^{th}$ robot's travelling time is $k_i^{-1}$) , $t_{l_i}$ and $t_{u_i}$ are lower and upper bounds of the travel time of the $i^{th}$ robot. The $i^{th}$ robot cannot reach its target sooner than $t_{l_i}$ because of its speed limitation, and cannot reach its target later than $t_{u_i}$ based on the problem definition. We find the velocity of the $i^{\text{th}}$ robot using the time derivative of (5) as

$$\boldsymbol{v}_i = \dot{\gamma}_i \frac{\partial \boldsymbol{r}_i}{\partial \gamma_i} = k_i \sum_{d=1}^{m} \left( d\alpha_{d,i} \boldsymbol{i} + d\beta_{d,i} \boldsymbol{j} \right) \gamma_i^{d-1} \tag{8}$$

As can be seen from (8), the velocity of each robot is a function of $k_i$ and we can adjust the speed of the robots using this parameter without changing the robots' paths.

In order to solve the conflict resolution problem of MMRs, instead of checking the safety constraints among robots constantly, which is computationally expensive, we first find all the potential collision zones and then find collision-free motion conditions.

After finding the paths of the robots, there are some intersections in the environment, referred to as potential collision zones. Figure 3a shows potential collision zones for five robots as an example, in which the potential collision zone between the $i^{\text{th}}$ and $j^{\text{th}}$ robot is denoted by $c_{ij}$. For collision-free motion, we need to ensure all the robots maintain a safe distance amongst each other only in potential collision zones. To find potential collision zones, we find places in the environment where the Euclidean distance between the $i^{\text{th}}$ and $j^{\text{th}}$ robot is shorter than a safety radius, as shown in Fig. 3b, i.e.,

$$\|\boldsymbol{r}_i(\gamma_i) - \boldsymbol{r}_j(\gamma_j)\| \leq \rho_{ij} \;\; i,j = 1,...,n \text{ and } i \neq j \tag{9}$$

where $\boldsymbol{r}_i$ and $\boldsymbol{r}_j$ designate the paths of the $i^{\text{th}}$ and $j^{\text{th}}$ robots, respectively, and $\rho_{ij}$ is the safety radius between them. The safety radius is determined based on the robots' size. We usually assign a value between one to three times of the summation of the $i^{th}$ and $j^{th}$ robots' bases to $\rho_{ij}$.

We assume the potential collision zone, $c_{ij}$, is a circle with a center located at $\boldsymbol{r}_{ij}$, which is the intersection of $\boldsymbol{r}_i$ and $\boldsymbol{r}_j$, as shown in Fig 3b. This circle represents the area where the $i^{\text{th}}$ and $j^{\text{th}}$ robots may collide (please see (9)). To identify $c_{ij}$, we need to find $\boldsymbol{r}_{ij}$. This intersection point ($\boldsymbol{r}_{ij}$) can be found from $\|\boldsymbol{r}_i - \boldsymbol{r}_j\| = 0$; however in some cases, even though the intersection point does not exist, the robots do not satisfy the safety constraints, e.g., $c_{15}$ in Fig. 3a. First, we obtain the borderlines of the collision zone (beyond which the safety constraint is satisfied). These borderlines are obtained by $\|\boldsymbol{r}_i - \boldsymbol{r}_j\| = \rho_{ij}$ and finding a set of solutions: $\{(\boldsymbol{r}^j_{i_1}, \boldsymbol{r}^i_{j_1}), (\boldsymbol{r}^j_{i_2}, \boldsymbol{r}^i_{j_2})\}$. Second, to obtain $\boldsymbol{r}_{ij}$, we assume that robots are moving on straight lines inside $c_{ij}$. This assumption will not affect the position of $r_{ij}$ significantly. In addition, a safety margin will be considered in assigning the safety radius $\rho_{ij}$ to compensate this assumption. Thus, $\boldsymbol{r}_{ij}$ is simply the intersection of two straight lines connecting $\boldsymbol{r}^j_{i_1}$ to $\boldsymbol{r}^i_{j_1}$ and $\boldsymbol{r}^j_{i_2}$ to $\boldsymbol{r}^i_{j_2}$. Third, we assign the maximum of the distances between $\boldsymbol{r}_{ij}$ and the points in $\{\boldsymbol{r}^j_{i_1}, \boldsymbol{r}^i_{j_1}, \boldsymbol{r}^j_{i_2}, \boldsymbol{r}^i_{j_2}\}$ as the radius of $c_{ij}$. Note that circle $c_{ij}$ is the circumscribed circle of the solution points with a center of $\boldsymbol{r}_{ij}$.

After finding the potential collision zones, we propose a condition for collision-free motion. To find this condition, we relate $\gamma$s of the $i^{\text{th}}$ and $j^{\text{th}}$ robots to each other in a potential collision zone $c_{ij}$. Therefore, we find collision $\gamma$s, $\gamma_i^j$ and $\gamma_j^i$ of the $i^{\text{th}}$ and $j^{\text{th}}$ robots, respectively, where

$$\boldsymbol{r}_{ij} \simeq \boldsymbol{r}_i(\gamma_i^j) \simeq \boldsymbol{r}_j(\gamma_j^i) \tag{10}$$

As mentioned, the potential collision point $\boldsymbol{r}_{ij}$ does not always lie on the paths $\boldsymbol{r}_i$ and $\boldsymbol{r}_j$. Consequently, $\boldsymbol{r}_{ij}$, $\boldsymbol{r}_i(\gamma_i^j)$, and $\boldsymbol{r}_j(\gamma_j^i)$ are not necessarily the same point, although they are very close. By having a set of solutions $\{\gamma_i^j, \gamma_j^i\}$, we relate $\gamma_i^j$ to $\gamma_j^i$ by assuming a collision is occurring at time $t_{ij}$ on $\boldsymbol{r}_{ij}$. From (7), we have

$$t_{ij} = \gamma_i^j / k_i = \gamma_j^i / k_j \tag{11}$$

Therefore, for the potential collision zone $c_{ij}$ using (9), (11) and

$$\gamma_j^i = \gamma_i^j k_j / k_i \tag{12}$$

we obtain a condition for collision-free movement in the potential collision zone $c_{ij}$ as follows:

$$\begin{aligned} \|\boldsymbol{r}_i(\gamma_i^j) - \boldsymbol{r}_j(\gamma_i^j k_j / k_i)\| &> \rho_{ij} \\ \forall i,j \in \{1,...,n\} &\text{ and } i \neq j \end{aligned} \tag{13}$$

where the condition expressed in (13) is a sufficient condition for collision-free movement of robots $i$ and $j$. This condition implies that only if the distance between the robots is larger than their safety radius in the potential collision zones, there will be no collision in the environment. Therefore, to guarantee collision-free movement, (13) must be satisfied for all the potential collision zones. The non-equality constraints given by (13) constitute the conflict resolution for a MMR system.

In the solutions set of potential collision points, there are some solutions which lie in the same potential collision zone, i.e., there are duplicates of the zones. To find the unique number of potential collision zones, we use the K-means clustering algorithm. For every two robots, we initially find $n_p$ pairs of potential collision points, $(\gamma_i^j, \gamma_j^i)$. We then use Algorithm 1 to cluster the solutions and find a unique number of potential collision zones with no overlap.

**Algorithm 1** Clustering Algorithm for Finding Unique Potential Collision Zones: Cluster($\gamma_i^j, \gamma_j^i$)

1: let $c = 1$, $D_c = \emptyset$
2: **while** $\max D_c < a$ and $c \leq n_p$ **do**
3: Classify pairs of $(\gamma_i^j, \gamma_j^i)$ to $c$ clusters using K-means clustering algorithm
4: $c \Leftarrow c + 1$
5: **end while**

where $a$ is the maximum acceptable distance of the points in each collision zone's cluster and is a small value $\sim 0.1$. Large $a$ results in more overlap on the potential collision zones and ultimately more duplicates. $D_c$ is the set of distances between each cluster point to the center of the cluster, and $n_p$ is the initial number of solutions of potential collision points, i.e., (10) for all $i,j \in \{1,...,n\}$, found a priori in Section III-A.

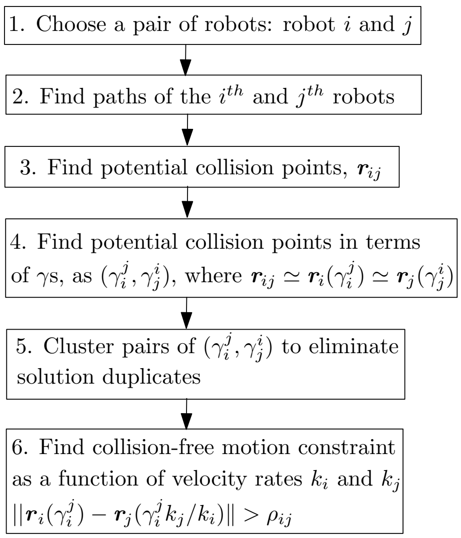



Fig. 4: Our proposed procedure for finding potential collision zones by coordinating robots' speeds.

A summary of our proposed strategy to find collision-free motion constraints is presented in Fig. 4. The given steps in the figure show that in order to find collision-free motion equations for robots $i$ and $j$, we must first find potential collision points in terms of $\gamma$s. Next, we use the clustering algorithm to eliminate the duplicates of these potential collision zones. At the end, by having collision points in terms of $\gamma$s, we propose collision-free motion as a function of the velocity rates $k_i$ and $k_j$. In the next step of conflict resolution, we find optimal velocity rates for all the robots in an optimization problem, which will be explained in the next section.

### *B. Motion-liveness*

Motion-liveness enables MMRs to reach their targets without stopping while in movement, resulting in smooth motion of MMRs. Therefore, our attempt is to avoid any collision in the system and maximize the speed of the robots. To do so, we construct an optimization problem where the objective is to maximize the speeds of the robots on smooth paths, as discussed earlier, while avoiding conflicts. Maximizing the speed of the robots satisfies motion-liveness in addition to getting the robots to reach their targets in a short amount of time. To achieve this, it is sufficient to maximize $k_i$s, i.e., minimizing robots' traveling time $k_i^{-1}$, subject to collision-free motion condition expressed in (13)

$$\underset{k_1,\ldots,k_n}{\text{minimize}} \sum_{i=1}^{n} {k_i}^{-1} \tag{14}$$

$$\text{subject to: } \|\boldsymbol{r}_i(\gamma_i^j) - \boldsymbol{r}_j(\gamma_i^j k_j/k_i)\| > \rho_{ij} \tag{15}$$

$$t_{u_i}^{-1} \le k_i \le t_{l_i}^{-1} \tag{16}$$

$$i, j = 1, ..., n \;\; i \neq j$$

where $t_{u_i}$ and $t_{l_i}$ are explained in (7), and $\gamma_i^j$ and $\gamma_j^i$ are found in Section III-A. Solving an optimization problem with non-linear constraints is more computationally expensive compared to a linear constrained problem, especially for large-scale problems. We convert this optimization problem into a new one by enforcing the non-linear constraints stated in (13) to the objective function with a penalty factor of $\lambda$

$$\underset{k_1,\ldots,k_n}{\text{minimize}} \sum_{i=1}^{n} \left( {k_i}^{-1} + \lambda \sum_{j=i}^{n} f_{ij}^2 U(f_{ij}) \right) \tag{17}$$

$$\text{subject to: (16)}$$

where $U$ is the step function, $\lambda$ is the penalty factor for the collision-free movement constraints and is a large number ($\simeq 10^{15}$), and

$$f_{ij} \;\; \equiv \;\; \rho_{ij} - \|\boldsymbol{r}_i(\gamma_i^j) - \boldsymbol{r}_j(\gamma_i^j k_j/k_i)\| \tag{18}$$

When the constraints stated in (13) are not satisfied, i.e., $f_{ij} \ge 0$, $U(f_{ij}) = 1$, the objective value (given by (17)) becomes large and the solution becomes infeasible. When the constraints are satisfied, i.e., $f_{ij} < 0$, then $U(f_{ij}) = 0$ and the solution becomes feasible. In summary, if (17) has a feasible solution, collision-free motion with motion-liveness is guaranteed.

Since our optimization problem is highly constrained and multi-modal, one of the most efficient methods for solving it is to use meta-heuristic algorithms. We use three of the well-known meta-heuristic optimization methods: Genetic Algorithm (GA) [49], Particle Swarm Optimization (PSO) [50], and Simulated Annealing (SA) [51].

In dynamic environments with moving obstacles, a strategy is required to generate collision-free motion while performing obstacle avoidance. A sample of a dynamic environment and potential collision zones are illustrated in Fig. 5. When there are moving obstacles in the environment, the robots' paths have to be updated to avoid obstacle collision (path re-planning). Consequently, some new potential collision zones will be created in the environment. The dotted paths in Fig. 5 show the robots' paths before the obstacles have moved, and solid lines illustrate the updated paths. For these dynamic environments, on-line conflict resolution is required to overcome new potential collision zones (blue circles in Fig. 5) which result from path re-planning.

In our strategy of conflict resolution for dynamic environments, we find $\gamma$s where the paths of the robots collide with obstacles. Next, we perform the following steps: **a)** resolve conflicts before facing new obstacles, **b)** re-plan the robot paths to avoid the obstacles, **c)** find the new potential collision zones, and **d)** resolve new conflicts according to (17). Algorithm 2 shows the details of our proposed dynamic conflict resolution.

### *C. An Innovative Algorithm for Fast Conflict Resolution*

The conflict resolution problem in cluttered environments is computationally expensive. The computational cost increases exponentially with an increase in the number of robots. In real-time applications, having fast conflict resolution is necessary. We propose a novel algorithm of Environment Clustering (EC) that can resolve the conflicts for cluttered environments with very low computational cost.

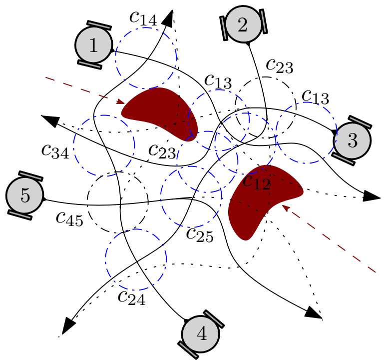



Fig. 5: Potential collision zones in a dynamic environment with moving obstacles.

---

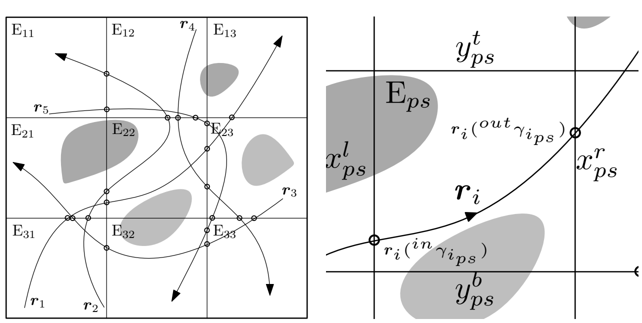
**Algorithm 2** Conflict Resolution for Dynamic Environments

---

**Input:** $\boldsymbol{I}_o$
**Output:** $k_1, ..., k_n$
  Perform conflict resolution: find $k_1, ..., k_n$ by solving (17)
  **for** robots $i = 1, ..., n$ **do**
    Solve $\min \|\boldsymbol{r}_i(\gamma_i^o) - \boldsymbol{I}_o\| = \rho_{ij}$ for $\gamma_i^o \in [0, 1]$
    Find collision time: $t_i^o = \gamma_i^o / k_i$
    Path re-planning $\boldsymbol{r}_i$ (using (4)) to bypass the $o^{th}$ obstacle, on $\gamma_i \in [\gamma_i^o, 1]$
  **end for**
  Find $h$ : the index of the first robot that collides with the obstacle: $t_h^o = \min\{t_1^o, ..., t_n^o\}$
  Perform conflict resolution on $\gamma_i \in [k_i t_h^o, 1], \forall i \in \{1, ..., n\}$: find $k_1, ..., k_n$ by solving (17)

where $\boldsymbol{I}_o$ is the set of border coordinates of the $o^{th}$ obstacle, $\gamma_i^o$ is $\gamma_i$ when the $i^{\text{th}}$ robot collides the obstacle, $t_i^o$ is time when $i^{\text{th}}$ robot collides with the obstacle, and $h$ is the index of the first robot that collides with the obstacle.

---




(a) Division of robots' paths in environment clustering.

(b) Division of the $i^{th}$ robot's path in environment cluster $\mathrm{E}_{ps}$.

Fig. 6: Environment clustering.

Our proposed method is to cluster the environment into smaller environments and resolve the conflicts of each cluster separately. The procedure for this clustering algorithm is as follows:

- First, we divide the environment into $p \times s$ smaller environments. As an example, $3 \times 3$ environment clusters are shown in Fig. 6a. The parameters $p$ and $s$ are determined depending on the environment complexity and required accuracy[2].
- Second, for each robot $\boldsymbol{r}_i$, we find $[^{in}\gamma_i, ^{out}\gamma_i)_{ps}$ in $\boldsymbol{r}_i$ which are the boundaries of $\gamma_i$ of the $i^{th}$ robot in environment cluster $\mathrm{E}_{ps}$. These boundaries specify the path section of the $i^{th}$ robot in $\mathrm{E}_{ps}$ as illustrated in Fig. 6b, i.e., when $\gamma_i \in [^{in}\gamma_i, ^{out}\gamma_i)_{ps}$ the $i^{th}$ robot is in $\mathrm{E}_{ps}$. To find the boundaries of $\gamma_i$ in $\mathrm{E}_{ps}$ we find the points $\boldsymbol{r}_i\left(^{in}\gamma_{i_{ps}}\right)$ and $\boldsymbol{r}_i\left(^{out}\gamma_{i_{ps}}\right)$ where the borders of $\mathrm{E}_{ps}$ intersect $\boldsymbol{r}_i$ as shown in Fig 6b.
- Third, we find potential collision zones' boundaries in each environment cluster $\mathrm{E}_{ps}$ as
$$\begin{aligned} \|\boldsymbol{r}_i(\gamma_i) - \boldsymbol{r}_j(\gamma_j)\| &= \rho_{ij} \\ \gamma_i &\in [^{in}\gamma_i, ^{out}\gamma_i)_{ps} \\ \gamma_j &\in [^{in}\gamma_j, ^{out}\gamma_j)_{ps} \end{aligned} \tag{19}$$
For each individual environment, we subsequently find the potential collision zones constraints of collision-free motion given by (13).
- Forth, we execute conflict resolution on each independent environment.

The main advantage of using this approach is that parallel programming techniques can be exploited, thus reducing execution time.

## D. MMRs Motion Constraints

This section presents the procedure of generating feasible paths for the robots considering velocity and non-holonomic constraints. Based on the velocity equation found in (8), the components of the velocity in $x$ and $y$ directions are

$$\dot{x}_i = k_i \sum_{d=1}^{m} \left(d\alpha_{di}\gamma_i^{d-1}\right), \; \dot{y}_i = k_i \sum_{d=1}^{m} \left(d\beta_{di}\gamma_i^{d-1}\right) \tag{20}$$

The maximum velocity of the robot is

$$\max_{\gamma\in[0,1]} \upsilon_i = \max_{\gamma\in[0,1]} \sqrt{\dot{x}^2 + \dot{y}^2} \leq R_i\Omega_i \tag{21}$$

where $R_i$ is the wheel radius of the $i^{\text{th}}$ robot, and $\Omega_i$ is the maximum angular velocity of the $i^{\text{th}}$ robot's wheels. Then from (20) with the given upper bound for $k_i \leq t_{l_i}^{-1}$ from (7), we rewrite (21) as

$$\max_{\gamma\in[0,1]} \sqrt{\left(\sum_{d=1}^{m}\left(d\alpha_{di}\gamma_i^{d-1}\right)\right)^2 + \left(\sum_{d=1}^{m}\left(d\beta_{di}\gamma_i^{d-1}\right)\right)^2} \leq R_i\Omega_i t_{l_i} \tag{22}$$

[2] Although increasing the number of clusters will improve the computation efficiency, based on our observations, increasing the number of clusters after a certain number (e.g., more than 4 by 4 in our problem) results in potential collision zones overlap among the clusters and duplicates of the zones. It is obvious that duplicates of potential collision zones increase the computation.

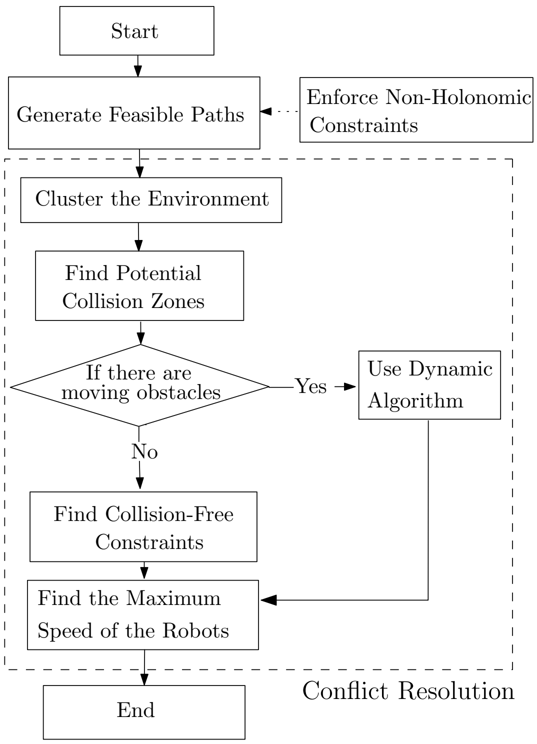



Fig. 7: Proposed conflict resolution method for cluttered environments.

To find the $\boldsymbol{r}_i$'s coefficients $\alpha_{0i}, ..., \alpha_{mi}, \beta_{0i}, ..., \beta_{mi}$, we fit (5) on the desired path points, $\boldsymbol{P}_i$, subject to the velocity constraints over the robots' paths. Spline fitting is an optimization problem that minimizes the distance between the path $\boldsymbol{r}_i$ and the desired points $\boldsymbol{P}_i$ of the $i^{\text{th}}$ robot

$$\underset{\alpha_{0,i},\ldots,\alpha_{m,i},\beta_{0,i},\ldots,\beta_{m,i}}{\text{minimize}} \quad ||\boldsymbol{r}_i - \boldsymbol{P}_i|| + \max_{\gamma_i} g(\boldsymbol{r}_i) \tag{23}$$

subject to: (22)

The desired orientation of the robot is found by the non-holonomic constraint as

$$\theta_i = \tan^{-1}(\dot{y}_i/\dot{x}_i) \tag{24}$$

Equations (23) and (24) give the desired position and orientation of the $i^{th}$ robot, which satisfy both velocity and non-holonomic constraints and will be used for robot control using the controller developed in [52].

Our complete conflict resolution method is shown in the given flow chart in Fig. 7. We first generate the robots' paths by enforcing the robots' and environments' constraints. Thereafter, we execute the conflict resolution algorithm, which is as follows: cluster the environment and subsequently find the potential collision zones. Using these collision zones, the algorithm determines if there are moving obstacles in the environment. If so, the dynamic algorithm will be performed, otherwise, the collision-free constraints and the robots' maximum speeds are found, and will be sent to the robots.

TABLE I: Optimization parameters used for conflict resolution.

| GA | | PSO | | SA | |
|---|---|---|---|---|---|
| Population size | 150 | Population size | 100 | Initial temperature | 1000 |
| Crossover rate | 0.85 | Cognitive attraction | 0.6 | Reannealing interval | 200 |
| Mutation rate | 0.15 | Social attraction | 1.25 | | |
| Pareto fraction | 0.3 | Generations | 1000 | | |
| Generations | 30000 | | | | |

TABLE II: Run time of conflict resolution using the environment clustering algorithm.

| | | Run Time (sec) | |
|---|---|---|---|
| **Number of robots** | **Number of potential Collision zones** | **No environment clustering, (SA)** | **Clustering to $3 \times 3$ environments, (SA-EC)** |
| 20 | 196 | 0.92 | 0.63 |
| 60 | 1974 | 1.12 | 0.87 |
| 100 | 4357 | 1.46 | 1.23 |

## IV. Results and Discussion

### A. Simulation Results

This section presents the simulation results using different metaheuristic optimization techniques, GA, PSO and SA, conflict resolution of a dynamic environment, and conflict resolution of cluttered environment using Environment Clustering (EC).

MATLAB program on Intel (R) Core$^{\text{TM}}$ i5-4690K with 3.50 GHz CPU is used to run the optimization algorithms for conflict resolution. Table I shows the parameters used in the implementation of the optimization algorithms.

*1) A Large Number of Robots in Cluttered Environments:* We discuss the conflict resolution results in two perspectives: (i) computational cost, and (ii) performance. In the first case, we explore the run time for different algorithms which takes to find collision-free motion. In the second case, we discuss the performance of these optimization algorithms in finding maximum velocities of MMRs moving collision-free. An example of a cluttered environment can be seen in Fig. 10.

For case (i), computational cost perspective, results for different numbers of robots using different optimization algorithms are shown in Fig. 8. The axis on top of this

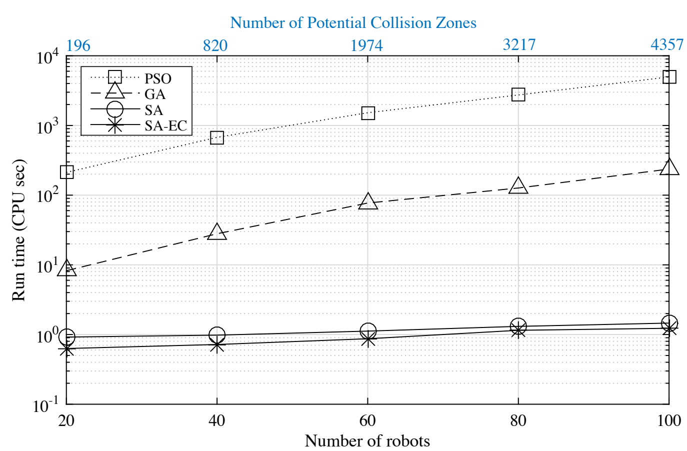



Fig. 8: Run time of conflict resolution for cluttered environments using different optimization methods.

figure denotes the number of potential collision zones in the environment (e.g., for 60 robots the environment has 1974 potential collision zones). As can be seen the algorithm is capable of solving the problem even for 100 of robots where there are 4357 potential collision zones. We observed that SA is the fastest method for conflict resolution and finds collision-free movements of 100 robots in 1.46 seconds. GA is slightly slower than SA, but it finds the solution in 4 minutes while PSO takes 40 minutes. We also improved the run time by combining SA with EC, which its results are discussed in Section IV-A3.

For case (ii), performance perspective, as Fig. 9 shows GA finds better solution which means robots have higher speeds than the solutions found by SA and PSO. To compare the performance of these algorithms, we introduce a performance index called the average traveling time of the robots. We plotted this index for the three optimization algorithms in Fig. 9. GA resulted in a faster motion of the robots compared to SA and PSO.

*2) Dynamic Environments:* For dynamic environments with moving obstacles, we implemented our conflict resolution algorithm (Algorithm 2). Figure 10 illustrates a sample dynamic environment with 30 robots and four obstacles. First, as a result of conflict resolution, robots begin moving collision-free, then, the obstacles enters causing all the robots' that are in contact with the obstacle re-plan their paths (updated paths are shown in red dotted lines in Fig. 10b). As a result of re-planning, new potential collision zones are created, and then using SA we found collision-free motion in 0.35 seconds.

*3) Environment Clustering:* The computational times of conflict resolution for three different numbers of robots with Simulated Annealing and Environment Clustering (SA-EC) are shown in Fig. 8. We clustered the environment to $3 \times 3$ environments. It is evident that the environment clustering algorithm enables the conflict resolution problem to be solved more quickly, 100 of robots in less than 1.23 seconds. The computation time is decreased by up to 32% with SA-EC as shown in Table II.

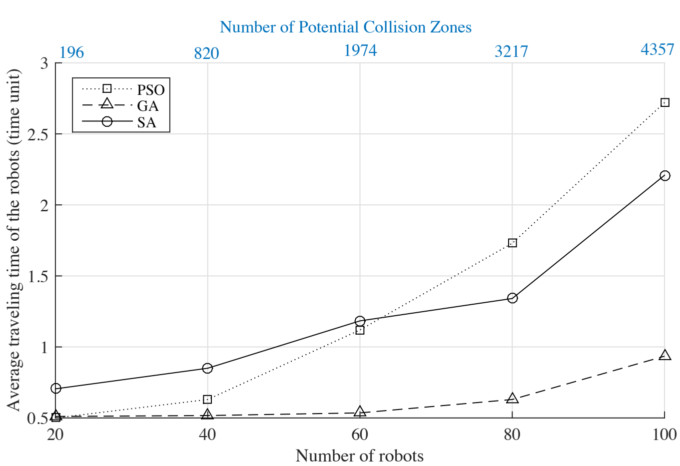
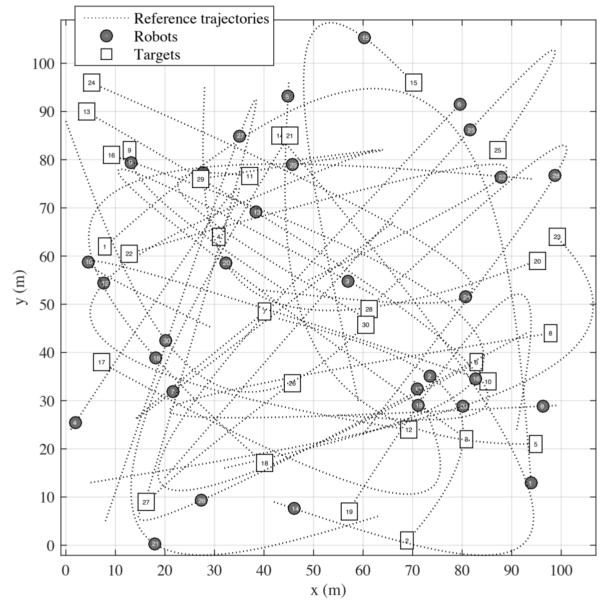
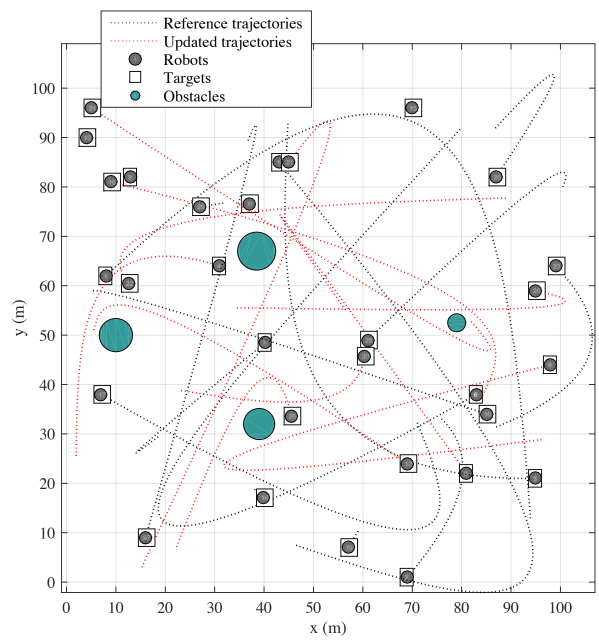



Fig. 9: Average traveling time of the robots using different optimization methods.




(a) Conflict resolution in the environment before obstacles enter.




(b) Path re-planning to avoid obstacles collision.

Fig. 10: Results of conflict resolution in a dynamic environment.

## B. Experimental Results

This section presents the implementation of conflict resolution in an experimental setup of five non-holonomic mobile robots. We developed a testbed to verify our methodologies experimentally as well. Fig. 11 shows the diagram of our experimental setup. We used five Arduino mobile robots equipped with wireless Xbee modules for communication among them and a computer. For measuring the robots' position and orientation, we used VICON system. The computed outputs of our developed algorithm were sent to the robots

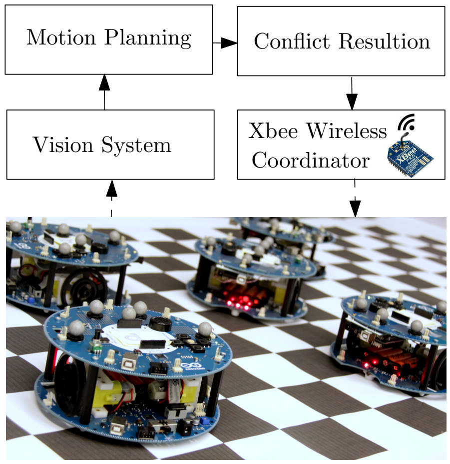



Fig. 11: Experimental setup of multiple-mobile robots.

TABLE III: Experiments: parameters and results.

| Parameter | Value |
|---|---|
| *Parameters used in experiment:* | |
| $n$ number of robots | 5 |
| Number of potential collision zones | 5 |
| safety radius $\rho_{ij}$ for each two robots | 30 $cm$ |
| $t_{l_i}$ | 2 $sec$ |
| $t_{u_i}$ | 10 $sec$ |
| *Experiment results:* | |
| Conflict resolution run time | 0.2 $sec$ |
| Objective function value | 20 |
| Minimum distance among the robots, measured during the experiment | 40 $cm$ |
| Average traveling time of the robots | 4 $sec$ |

TABLE IV: Comparisons of different approaches reported in the literature.

| | Simulation run time ($s$) | Processor base frequency (GHz) | Scalability | Motion-liveness | Cluttered Environment | Dynamic environment | Number of robots for experimental verifications |
|---|---|---|---|---|---|---|---|
| [28] | ?[a] | | ✓ | – | ✓ | – | – |
| [47] | ?[a] | | – | – | – | ✓ | – |
| [37] | ?[a] | | ✓ | ✓ | – | – | – |
| [31] | ?[a] | | ?[a] | ✓ | – | – | – |
| [16] | **10** for 60 robots | 2.3 | ✓ | – | ✓ | – | – |
| [33] | **0.38** for 30 robots | 2.4 | ✓ | – | – | – | – |
| [40] | ?[a] | | ✓ | – | ?[a] | ✓ | 5 |
| [42] | ?[a] | | – | ✓ | – | ✓ | 4 |
| [24] | **0.05** for 30 robots | 3 | ?[a] | – | ✓ | – | 2 |
| [27] | **20** for 8 robots | 3.07 | – | ✓ | – | ✓ | 4 |
| [41] | **10**[b] for 100 robots | ?[a] | ✓ | ✓ | – | ✓ | 6 |
| [39] | **0.1**[b] for 100 robots | ?[a] | ✓ | – | – | – | 8 |
| [22] | **1.5** for 100 robots | 1.5 | ✓ | – | – | – | 3 |
| Our method | **1.23** for 100 robots[c] | 3.50 | ✓ | ✓ | ✓ | ✓ | 5 |

[a]Not reported.

[b]The robots' targets were unlabeled and being assigned to avoid intersections in these works. It is applicable to tasks with unlabeled robots without individual targets. Therefore, they cannot be applied to this problem of conflict resolution.

[c]This is for a highly cluttered environment with 4357 potential collision zones (please see the results presented in Table II).

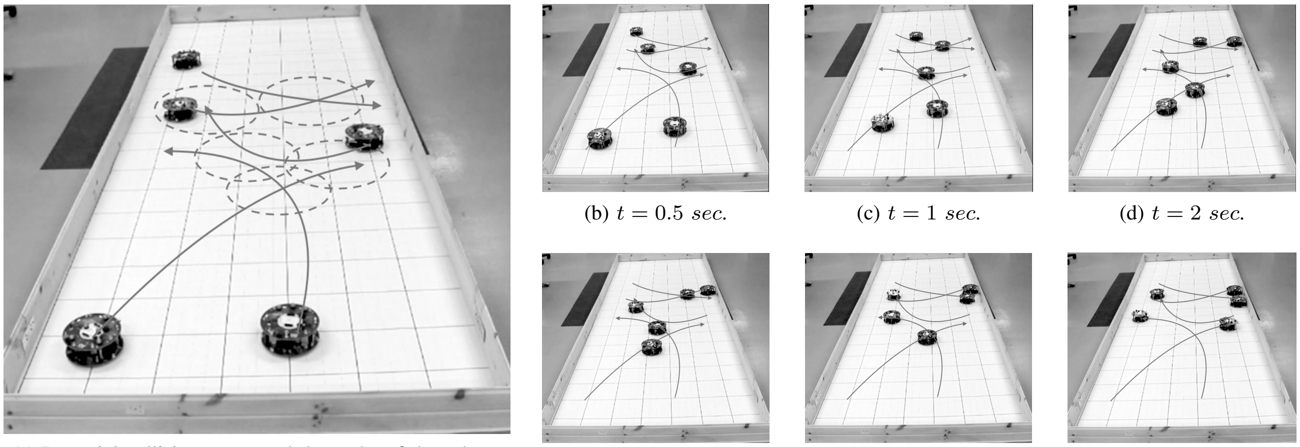
with wireless communications. The frequency of the sampling rate of the wireless communication between the computer and the modules (Xbee) is 100 Hz. The same frequency is used for feedback with VICON system. The maximum speed of robots is 0.65 $m/s$ with the wireless communication range of 100 $m$.

Our environment, consisting of five robots with five collision zones, is shown in Fig. 12. The robots' paths which were found based on the motion constraints explained in Section III-D, are also shown in this figure. The safety radius among the robots was 30 cm. Since SA was the fastest algorithm in the simulations, we used it to perform the experiments. The results showed that the algorithm can be successfully implemented in real-time. The SA algorithm finds collision-free motion of robots in 0.3 seconds.

Table IV presents the advantages of our conflict resolution approach in comparison to other works reported in the literature. While some methods such as [16], [33] reported scalability with a reasonable run time, very few considered live motions [41], [42] or incorporated cluttered environments [16], [24], [28]. Compared with [22], our method resolves MMRs conflicts in cluttered environments with a significantly higher number of intersections within a short run time. Although, cluttered environments were addressed in [24], our method proposed a new perspective to conflict resolution which is scalable and guarantees robots motion-liveness. The run time of [39] is less than other works; however, it should be noted that [39] deals with a task assignment problem, which has a different objective than ours. Using our developed strategy we showed that one can generate collision-free motion for a large number of robots - with thousands of intersections in the robots' paths- within only a few seconds (e.g., it takes less than a second to find collision-free motion for 60 robots with around 2000 path intersections, see Fig. 8). Our methodology can solve the main themes of conflict resolution in terms of scalability, motion-liveness, effectiveness in cluttered and dynamic environments.

## V. Conclusion

In this paper, we developed a new conflict resolution method that guarantees motion-liveness for a large number of robots in cluttered environments. We presented a mathematical formulation for conflict resolution and constructed it as an optimization problem that can be solved using meta-heuristic optimization techniques. We used Genetic Algorithm (GA), Particle Swarm Optimization (PSO), and Simulated Annealing (SA). Our extensive simulation results showed that our approach is capable of solving conflict resolution problems especially for a large number of robots in cluttered and dynamic environments. The simulation results for different numbers of robots showed that SA gives the best results when compared to GA and PSO. SA finds the collision-free motion for 100 robots in less than 1.23 seconds where there are 4357 potential collision zones (robots' path intersections) using an Intel core-i5 3.5 GHz CPU while other methods took significantly longer. Conflict resolution using GA resulted in a faster motion of the robots in the cluttered environment; around 60% faster motion



(a) Potential collision-zones and the paths of the robots.
(b) $t = 0.5\ sec.$
(c) $t = 1\ sec.$
(d) $t = 2\ sec.$
(e) $t = 3\ sec.$
(f) $t = 4\ sec.$
(g) $t = 5.5\ sec.$

Fig. 12: Experimental results of five robots conflict resolution.

than results obtained by using SA and PSO. This algorithm can be easily implemented in on-line conflict resolution while avoiding obstacles. We proposed an innovative environment clustering technique for real-time implementations as well. By clustering the environment, we reduced the execution time down by 32%. Using this clustering algorithm, we divided the conflict resolution problem into independent problems where parallel programming techniques can be exploited. We also validated our proposed methods using an experimental testbed that we developed. The experimental results verified that our approach is applicable to real-time applications. While the experiments were limited to 5 robots, simulations provide evidence that our strategy can also be adapted for large numbers as well. We compared our approach with other methods in the literature and showed not only our approach can solve the main themes of conflict resolution in terms of scalability, motion-liveness, effectiveness, it is readily applicable in cluttered and dynamic environments.

Our future work is focused on considering the dynamics of the mobile robots in solving the conflict resolution problem. This formulation can also easily be extended from planar to aerial vehicles.

## Acknowledgment

We would like to thank the Natural Sciences and Engineering Research Council of Canada (NSERC) to fund our research through their Discovery Grant program.

## Appendix A

This appendix presents a formulation of collision-free motion constraint by using a quadratic $\gamma_i$. We define a quadratic $\gamma_i$ function for the $i^{th}$ robot as $\gamma_i = k_{i,1}t^2 + k_{i,2}t$, where $k_{i,1}$ and $k_{i,2}$ are $\gamma_i$'s coefficients, and $t$ is time. By following the same procedure defined in the paper, we derive a collision-free motion constraint. Assume a collision at $\boldsymbol{r}_i(\gamma_i(t_{ij})) \simeq \boldsymbol{r}_j(\gamma_j(t_{ij})) \simeq \boldsymbol{r}_{ij}$, we relate collision $\gamma$s, $\gamma_i^j = \gamma_i(t_{ij})$ and $\gamma_j^i = \gamma_j(t_{ij})$ of the $i^{th}$ and the $j^{th}$ robots at the collision point $\boldsymbol{r}_{ij}$, respectively,

$$\gamma_i^j = k_{i,1}t_{ij}^2 + k_{i,2}t_{ij} \tag{25}$$

$$\gamma_j^i = k_{j,1}t_{ij}^2 + k_{j,2}t_{ij} \tag{26}$$

By finding collision time from equations (25) and (26), we write the following equation to relate $\gamma_i^j$ and $\gamma_j^i$ at the collision time

$$t_{ij} = \frac{-k_{i,2} \pm \sqrt{k_{i,2}^2 + 4k_{i,1}\gamma_i^j}}{2k_{i,1}} = \frac{-k_{j,2} \pm \sqrt{k_{j,2}^2 + 4k_{j,1}\gamma_j^i}}{2k_{j,1}} \tag{27}$$

Then, by solving (27), we find $\gamma_j^i$

$$\gamma_j^i = \frac{k_{i,2}^2 k_{j,1} - k_{i,1}k_{i,2}k_{j,1}^2 k_{j,2} + 2k_{i,1}k_{j,1}\gamma_i^j \pm \sigma_{ij}}{2k_{i,1}^2} \tag{28}$$

where $\sigma_{ij}$ is

$$\sigma_{ij} = \sqrt{\begin{aligned}&4\gamma_i^j k_{i,1}^3 k_{j,2}^2 + k_{i,1}^2 k_{i,2}^2 k_{j,1}^4 k_{j,2}^2 - 8\gamma_i^j k_{i,1}^2 k_{i,2} k_{j,1}^3 k_{j,2} \\ &-2k_{i,1}k_{i,2}^3 k_{j,1}^3 k_{j,2} + 4\gamma_i^j k_{i,1} k_{i,2}^2 k_{j,1}^2 + k_{i,2}^4 k_{j,1}^2\end{aligned}} \tag{29}$$

Thus, the collision-free motion of the $i^{th}$ and $j^{th}$ robots is

$$||\boldsymbol{r}_i(\gamma_i^j) - \boldsymbol{r_j}\left(\frac{k_{i,2}^2 k_{j,1} - k_{i,1}k_{i,2}k_{j,1}^2 k_{j,2} + 2k_{i,1}k_{j,1}\gamma_i^j \pm \sigma_{ij}}{2k_{i,1}^2}\right)|| > \rho_{ij}\ \forall i,j \in \{1,...,n\}, i \neq j \tag{30}$$

where $\sigma_{ij}$ is expressed in (29). This constraint is a function of $k_{i,1}, k_{i,2}, k_{j,1}$ and $k_{j,2}$, to be found for collision-free motion.

It can be easily seen that using this quadratic function of time will make the methodology complicated and computationally expensive.

## REFERENCES


[1] W. Hu, L. Liu, and G. Feng, "Consensus of linear multi-agent systems by distributed event-triggered strategy," *IEEE Transactions on Cybernetics*, vol. 46, pp. 148–157, Jan 2016.

[2] D.-H. Lee, S. Zaheer, and J.-H. Kim, "Ad hoc network-based task allocation with resource-aware cost generation for multirobot systems," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 12, pp. 6871–6881, 2014.

[3] H. I. Son, A. Franchi, L. Chuang, J. Kim, H. Bulthoff, and P. Giordano, "Human-centered design and evaluation of haptic cueing for teleoperation of multiple mobile robots," *IEEE Transactions on Cybernetics,*, vol. 43, pp. 597–609, April 2013.

[4] Y. Su and J. Huang, "Cooperative output regulation with application to multi-agent consensus under switching network," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 42, pp. 864–875, June 2012.

[5] M. U. Khan, S. Li, Q. Wang, and Z. Shao, "Cps oriented control design for networked surveillance robots with multiple physical constraints," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 35, pp. 778–791, May 2016.

[6] X. Wang, J. Qin, and C. Yu, "Iss method for coordination control of nonlinear dynamical agents under directed topology," *IEEE Transactions on Cybernetics*, vol. 44, pp. 1832–1845, Oct 2014.

[7] C. Yu, M. Zhang, F. Ren, and G. Tan, "Multiagent learning of coordination in loosely coupled multiagent systems," *IEEE Transactions on Cybernetics*, vol. 45, pp. 2853–2867, Dec 2015.

[8] C. Teulire, E. Marchand, and L. Eck, "3-d model-based tracking for uav indoor localization," *IEEE Transactions on Cybernetics*, vol. 45, pp. 869–879, May 2015.

[9] W. Li, "Notion of control-law module and modular framework of cooperative transportation using multiple nonholonomic robotic agents with physical rigid-formation-motion constraints," *IEEE Transactions on Cybernetics*, vol. 46, pp. 1242–1248, May 2016.

[10] S. Wen, W. Zheng, J. Zhu, X. Li, and S. Chen, "Elman fuzzy adaptive control for obstacle avoidance of mobile robots using hybrid force/position incorporation," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 42, pp. 603–608, July 2012.

[11] S. Hoshino and K. Maki, "Safe and efficient motion planning of multiple mobile robots based on artificial potential for human behavior and robot congestion," *Advanced Robotics*, vol. 29, no. 17, pp. 1095–1109, 2015.

[12] H. Poonawala, A. Satici, H. Eckert, and M. Spong, "Collision-free formation control with decentralized connectivity preservation for nonholonomic-wheeled mobile robots," *IEEE Transactions on Control of Network Systems*, vol. 2, no. 2, pp. 122–130, 2015.

[13] T.-T. Han and S. S. Ge, "Styled-velocity flocking of autonomous vehicles: A systematic design," *IEEE Transactions on Automatic Control*, vol. 60, no. 8, pp. 2015–2030, 2015.

[14] V. Digani, L. Sabattini, C. Secchi, and C. Fantuzzi, "Hierarchical traffic control for partially decentralized coordination of multi agv systems in industrial environments," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, (Hong Kong, China), pp. 6144–6149, May 2014.

[15] M. U. Khan, S. Li, Q. Wang, and Z. Shao, "Formation control and tracking for co-operative robots with non-holonomic constraints," *J. Intell. Robotics Syst.*, vol. 82, pp. 163–174, Apr. 2016.

[16] M. Cap, P. Novak, A. Kleiner, and M. Selecky, "Prioritized planning algorithms for trajectory coordination of multiple mobile robots," *IEEE Transactions on Automation Science and Engineering*, vol. 12, no. 3, pp. 835–849, 2015.

[17] M. Defoort and K. C. Veluvolu, "A motion planning framework with connectivity management for multiple cooperative robots," *Journal of Intelligent & Robotic Systems*, vol. 75, no. 2, pp. 343–357, 2014.

[18] Z. Yan, N. Jouandeau, and A. A. Cherif, "A survey and analysis of multi-robot coordination," *International Journal of Advanced Robotic Systems*, vol. 10, p. 399, 2013.

[19] M. Bristow, L. Fang, and K. W. Hipel, "Agent-based modeling of competitive and cooperative behavior under conflict," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 44, pp. 834–850, July 2014.

[20] G. Hollinger, S. Singh, *et al.*, "Multirobot coordination with periodic connectivity: Theory and experiments," *IEEE Transactions on Robotics*, vol. 28, no. 4, pp. 967–973, 2012.

[21] S. Kloder and S. Hutchinson, "Path planning for permutation-invariant multirobot formations," *IEEE Transactions on Robotics*, vol. 22, no. 4, pp. 650–665, 2006.

[22] M. Peasgood, C. M. Clark, and J. McPhee, "A complete and scalable strategy for coordinating multiple robots within roadmaps," *IEEE Transactions on Robotics*, vol. 24, no. 2, pp. 283–292, 2008.

[23] J. Chen and D. Sun, "Coalition-based approach to task allocation of multiple robots with resource constraints," *IEEE Transactions on Automation Science and Engineering*, vol. 9, no. 3, pp. 516–528, 2012.

[24] M. Hoy, A. S. Matveev, and A. V. Savkin, "Collision free cooperative navigation of multiple wheeled robots in unknown cluttered environments," *Robotics and Autonomous Systems*, vol. 60, no. 10, pp. 1253–1266, 2012.

[25] D. Zhu, H. Huang, and S. Yang, "Dynamic task assignment and path planning of multi-auv system based on an improved self-organizing map and velocity synthesis method in three-dimensional underwater workspace," *IEEE Transactions on Cybernetics*, vol. 43, pp. 504–514, April 2013.

[26] R. Cui, B. Gao, and J. Guo, "Pareto-optimal coordination of multiple robots with safety guarantees," *Autonomous Robots*, vol. 32, no. 3, pp. 189–205, 2012.

[27] D. Hennes, D. Claes, W. Meeussen, and K. Tuyls, "Multi-robot collision avoidance with localization uncertainty," in *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems-Volume 1*, pp. 147–154, International Foundation for Autonomous Agents and Multiagent Systems, 2012.

[28] J. Van Den Berg, S. J. Guy, M. Lin, and D. Manocha, "Reciprocal n-body collision avoidance," in *Robotics research*, pp. 3–19, Springer, 2011.

[29] L. W. Chen and P. C. Chou, "Big-cca: Beacon-less, infrastructure-less, and gps-less cooperative collision avoidance based on vehicular sensor networks," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. PP, no. 99, pp. 1–11, 2015.

[30] H.-T. Zhang, Z. Cheng, G. Chen, and C. Li, "Model predictive flocking control for second-order multi-agent systems with input constraints," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 62, no. 6, pp. 1599–1606, 2015.

[31] E. Roszkowska and S. Reveliotis, "A distributed protocol for motion coordination in free-range vehicular systems," *Automatica*, vol. 49, no. 6, pp. 1639–1653, 2013.

[32] F. Pasqualetti, A. Franchi, and F. Bullo, "On cooperative patrolling: Optimal trajectories, complexity analysis, and approximation algorithms," *IEEE Transactions on Robotics*, vol. 28, no. 3, pp. 592–606, 2012.

[33] A. Colombo and D. Del Vecchio, "Least restrictive supervisors for intersection collision avoidance: A scheduling approach," *IEEE Transactions on Automatic Control*, vol. 60, no. 6, pp. 1515–1527, 2015.

[34] L. Pallottino, V. G. Scordio, A. Bicchi, and E. Frazzoli, "Decentralized cooperative policy for conflict resolution in multivehicle systems," *IEEE Transactions on Robotics*, vol. 23, no. 6, pp. 1170–1183, 2007.

[35] Z.-H. Mao, D. Dugail, and E. Feron, "Space partition for conflict resolution of intersecting flows of mobile agents," *IEEE Transactions on Intelligent Transportation Systems*, vol. 8, no. 3, pp. 512–527, 2007.

[36] J. Alonso-Mora, T. Naegeli, R. Siegwart, and P. Beardsley, "Collision avoidance for aerial vehicles in multi-agent scenarios," *Autonomous Robots*, vol. 39, no. 1, pp. 101–121, 2015.

[37] S. A. Reveliotis and E. Roszkowska, "Conflict resolution in free-ranging multivehicle systems: A resource allocation paradigm," *IEEE Transactions on Robotics*, vol. 27, no. 2, pp. 283–296, 2011.

[38] Y. Ma, H. Wang, and M. Zamirian, "A novel approach for multiple mobile objects path planning: Parametrization method and conflict resolution strategy," *Physics Letters A*, vol. 376, no. 4, pp. 377–386, 2012.

[39] M. Turpin, N. Michael, and V. Kumar, "Capt: Concurrent assignment and planning of trajectories for multiple robots," *The International Journal of Robotics Research*, vol. 33, no. 1, pp. 98–112, 2014.

[40] E. Ferrera, A. R. Castano, J. Capitán, P. J. Marrón, and A. Ollero, "Multi-robot operation system with conflict resolution," in *ROBOT2013: First Iberian Robotics Conference*, pp. 407–419, Springer, 2014.

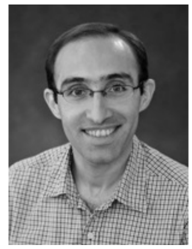
[41] M. Turpin, K. Mohta, N. Michael, and V. Kumar, "Goal assignment and trajectory planning for large teams of interchangeable robots," *Autonomous Robots*, vol. 37, no. 4, pp. 401–415, 2014.

[42] J. Snape, J. Van den Berg, S. J. Guy, and D. Manocha, "Smooth and collision-free navigation for multiple robots under differential-drive constraints," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Citeseer, 2010.

[43] A. Tuncer and M. Yildirim, "Dynamic path planning of mobile robots with improved genetic algorithm," *Computers & Electrical Engineering*, vol. 38, no. 6, pp. 1564–1572, 2012.

[44] R. Kala, "Multi-robot path planning using co-evolutionary genetic programming," *Expert Systems with Applications*, vol. 39, no. 3, pp. 3817–3831, 2012.
[45] Y. Zhang, D.-W. Gong, and J.-h. Zhang, "Robot path planning in uncertain environment using multi-objective particle swarm optimization," *Neurocomputing*, vol. 103, pp. 172–185, 2013.
[46] Q. Xu, "Collision avoidance strategy optimization based on danger immune algorithm," *Computers & Industrial Engineering*, vol. 76, pp. 268–279, 2014.
[47] J.-H. Liang and C.-H. Lee, "Efficient collision-free path-planning of multiple mobile robots system using efficient artificial bee colony algorithm," *Advances in Engineering Software*, vol. 79, pp. 47–56, 2015.
[48] R. Raja, A. Dutta, and K. Venkatesh, "New potential field method for rough terrain path planning using genetic algorithm for a 6-wheel rover," *Robotics and Autonomous Systems*, vol. 72, pp. 295–306, 2015.
[49] K. Deb, "An efficient constraint handling method for genetic algorithms," *Computer Methods in Applied Mechanics and Engineering*, vol. 186, no. 24, pp. 311 – 338, 2000.
[50] R. C. Eberhart and Y. Shi, "Particle swarm optimization: developments, applications and resources," in *Proceedings of the 2001 Congress on Evolutionary Computation, 2001.*, vol. 1, pp. 81–86, IEEE, 2001.
[51] R. Eglese, "Simulated annealing: a tool for operational research," *European journal of operational research*, vol. 46, no. 3, pp. 271–281, 1990.
[52] A. De Luca, G. Oriolo, and M. Vendittelli, "Control of wheeled mobile robots: An experimental overview," in *Ramsete*, pp. 181–226, Springer, 2001.

**Mohammadali Shahriari** (S15) received the B.Sc. degree in electrical engineering from the Iran University of Science and Technology, Tehran, Iran, in 2011 and the M.A.Sc. degree in mechatronics engineering from the Sharif University of Technology, Tehran, Iran, in 2013. He is currently working toward the Ph.D. degree in the School of Engineering, University of Guelph, Guelph, ON. His research interests include multiple mobile robots, intelligent control, optimization, and nonlinear control with applications to robotics systems.



**Mohammad Biglarbegian** (S07-M'??-SM??) received the B.Sc. degree (with honors) in mechanical engineering from the University of Tehran, Tehran, Iran, in 2002, the M.A.Sc. degree in mechanical engineering from the University of Toronto, Toronto, ON, and the Ph.D. degree in the Department of Mechanical and Mechatronics Engineering, University of Waterloo, Waterloo, ON. He is currently an Associate Professor with the School of Engineering Engineering, University of Guelph, Guelph, ON. His research interests include intelligent control, type-2 fuzzy logic systems and control, and nonlinear control with applications to mechatronics systems.