Time Optimal Rendezvous for Multi-Agent Systems Amidst Obstacles - Theory and Experiments

Bhaskar Vundurthy
Department of Electrical Engineering,
Indian Institute of Technology (IIT) Madras
Chennai, India
bhaskarvundurthy@gmail.com

K. Sridharan
Department of Electrical Engineering,
Indian Institute of Technology (IIT) Madras
Chennai, India
sridhara@iitm.ac.in

***Abstract*—Rendezvous of multiple autonomous agents has been of active interest in the last decade. Considerable work has been done on this problem with constraints on sensing and shape of the environment. However, not much is known on rendezvous amidst obstacles. When obstacles are introduced into the setting, it becomes natural to explore strategies for rendezvous that optimize some parameter (such as distance, time etc.). Our objective in this paper is to compute a location (which we refer to as the Time Optimal Rendezvous Point (TORP)) that minimizes the time for rendezvous amidst obstacles. We discuss challenges in finding TORP and develop efficient algorithms to compute TORP for r agents moving amidst m static polygonal obstacles. We then extend the analysis to handle dynamic obstacles. Experimental results are presented to validate the theory.**

***Index Terms*—Multi-agent systems, Time Optimal Rendezvous, Obstacles, Computational Geometry, Efficient Algorithms, Experiments**

I. Introduction

Distributed control of multi-agent systems [1] has been pursued actively during the last decade for accomplishing various tasks efficiently. Given the wide range of applications, it has attracted the attention of diverse research groups. In particular, multi-agent systems have become invaluable in handling security of cyber-physical energy systems [2], coupling heterogeneous cyber-physical production systems [3] and software methodologies [4].

A problem of interest with respect to multi-agent systems operating in a factory floor, hospital or other environments is *consensus*. Research on consensus includes strategies for rendezvous taking into account constraints on sensors and communication [5], [6]. Qin *et al.* [7] describe various contributions to consensus and coordination of multi-agent systems during the last decade.

In this paper, we consider a version of the classical rendezvous that has not received considerable attention in the literature. In particular, we address the rendezvous problem with obstacles introduced into the environment and compute a point that can be reached by the agents in minimum time from their (given) initial locations. We refer to this point as the Time Optimal Rendezvous Point (*TORP*) and denote it by R_t. We assume that each agent is a point mass (similar to assumptions in prior works) and further the starting locations of the agents are known to all the agents. Enforcing a constraint on the time can be related to the energy consumed by the system [8].

Having knowledge of the initial locations of the agents (as well as static obstacles) is practical in the following situations: (i) Groups of friendly agents that are employed in defense applications may be provided knowledge of the location of the others (in the group) and may want to meet to replenish or exchange their supplies (ii) Groups of agents engaged in emergency informatics may operate with knowledge of the individual locations so as to facilitate transfer of medical kits or other gadgets between them. Obstacles could be water bodies (or other entities) in the case of agents operating in outdoor environments while they could be chairs or desks (for example) in an indoor setting.

Optimization with respect to some criteria for the rendezvous problem has been considered in prior work. The authors in [9] present a solution to achieve rendezvous in minimum time for a network of first order agents with bounded inputs. A decentralized algorithm to calculate arrival angles at a precomputed TORP for Dubin's vehicles is reported in [10]. Recently, new techniques to compute TORP for multi-agent systems with velocity [11] and power [8] constraints have been reported. However, obstacles have not been considered in any of these formulations. The authors in [12] consider line of sight communication for a pair of agents and compute the point that minimizes the maximum distance amidst obstacles. An extension to r agents is reported in [13] for the same distance metric. However, time optimal rendezvous is not considered in [12] and [13]. Further, existing algorithms do not accommodate the presence of dynamic obstacles while applying the minimum time constraint.

We approach this problem by identifying that TORP is identical to the point in plane that minimizes the maximum time taken by any agent to arrive at the point. This leads to an algorithm to compute TORP for r agents. We then explore the location of TORP when it is computed on intermediate locations of agents, where each agent has traversed a finite time prior to arriving at these locations. We use these two results to compute the TORP for r agents moving amidst m polygonal static obstacles, followed by extending the results to handle dynamic obstacles.

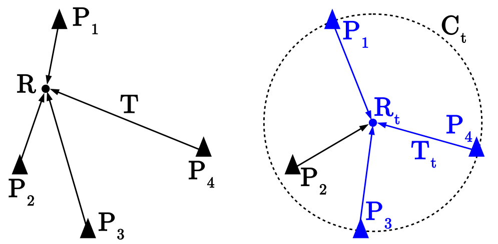
This work was supported in part by the Department of Science & Technology, Government of India through SERB project SB/S3/EECE/274/2016.

The rest of the paper is organized as follows. In section II, we compute the TORP for initial and intermediate locations of r agents in the absence of obstacles. Section III enhances the algorithm to handle m polygonal obstacles while section IV further enhances it to handle dynamic obstacles. Experimental results using multiple autonomous robotic agents in a constrained environment have been reported in Section V. Comparisons with prior works have been offered in section VI while section VII concludes the paper.

II. Algorithm for Computing TORP in the Absence of Obstacles

In this section, we present an algorithm to compute TORP for r agents in the absence of obstacles. We begin by formulating this problem as a minimax problem in travel times of agents. Such a formulation facilitates the computation of the TORP as the center of the smallest enclosing circle for all agent locations.

A. Computing TORP for r agents



(a) Rendezvous at R (b) Rendezvous at TORP, R_t

Fig. 1. Illustration of rendezvous points and travel times for four agents

The total time (T) for rendezvous at a location (say R) is equivalent to the time taken by all agents to arrive at the location. Without loss of generality, let the agents arrive at R in the order $P_1, P_2, \cdots, P_r$, indicating that P_1 arrives at R prior to every other agent while every other agent arrives at R before P_r. Thus the time taken for rendezvous is equal to the travel time of agent P_r which is the maximum time taken by any agent to arrive at R. Further, TORP (R_t) can then be computed by comparing the times for rendezvous at every possible location in the plane and identifying the point where the minimum occurs.

Fig. 1 illustrates the rendezvous of four agents at two distinct locations R and R_t. In Fig. 1(a), the agents arrive at R in the order P_1, P_2, P_3, P_4 and thus the time for rendezvous T is the time taken by P_4 to arrive at R. Fig. 1(b) hints at the computation of TORP. It can be observed that the rendezvous point R_t is equidistant to P_1, P_3 and P_4. Thus the time for rendezvous (denoted by T_t) is equal to the travel time of either of these agents. Further, the time for rendezvous at R_t (in Fig. 1(b)) is definitively lesser than the time for rendezvous at R (in Fig. 1(a)).

Consider the location of agent P_2 in Fig. 1(b). Since the time taken by P_2 to arrive at R_t is less than the time taken by remaining agents, agent P_2 does not have any affect on the location of R_t. In fact, as long as P_2 remains within the circle C_t, it would arrive at R_t prior to the remaining agents and thus cannot affect the location of TORP. It can further be observed that any addition of new agents within the circle C_t would not affect the location of TORP either. The following Theorem 1 utilizes these observations to compute the TORP.

Theorem 1: The TORP (R_t) for r identical agents is located at the center of the smallest enclosing circle that contains the initial locations of these r agents.

Proof: Without loss of generality, let m agents (where $m \in \mathbb{Z}^+, m < r$) lie on the smallest enclosing circle (denoted by C_t and centered at R_t) and the remaining $r - m$ agents lie within the circle C_t. The $r - m$ agents contained in the circle do not contribute to the TORP since their travel times to R_t are less than those of the m agents that lie on the circle C_t.

For the m agents that lie on the circle, consider a point P that is a finite distance away from R_t. The rendezvous time (T) taken by the m agents to arrive at P would be greater than the rendezvous time to arrive at the center of the circle R_t. Thus every other point in the circle can be discarded in lieu of the center of the circle as a candidate TORP. Consequently, the center of the circle R_t is indeed the TORP. **Q.E.D.**

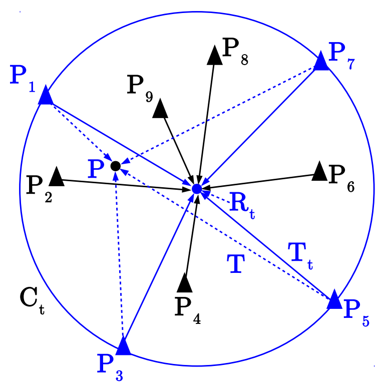


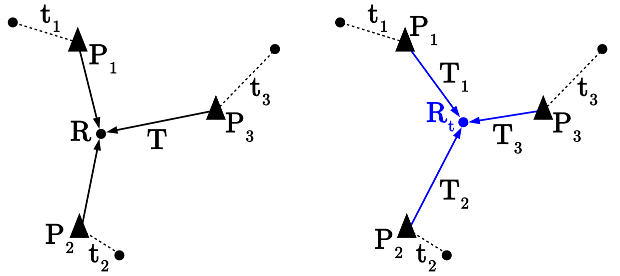
Fig. 2. TORP for 9 agents is the center of the smallest enclosing circle C_t

Fig. 2 illustrates Theorem 1 for 9 agents. Agents P_1, P_3, P_5, P_7 lie on the smallest enclosing circle ($m = 4$) while the remaining agents lie within the circle C_t. Due to their lower travel times to R_t, the remaining 5 agents (shown in black) do not affect its location. For an arbitrary rendezvous point P, agent P_5 takes the longest time (T) to arrive at P which is the rendezvous time for all agents. Since this time T is greater than the rendezvous time (T_t) to the center of the circle, the point P is ignored as a candidate TORP. With a similar analysis, every point in the plane has a longer rendezvous time compared to the center of the circle which proves that the TORP is indeed the center of the circle (Theorem 1).

In the following section, we extend this analysis to a scenario where the agents have elapsed a finite time before arriving at the locations that are used in computing the TORP.

B. Computing TORP using intermediate locations of r agents

In this section, the computation of R_t is performed on the locations of r agents given by $\{P_1, P_2, \cdots, P_r\}$, while taking into account the respective times (denoted by $\{t_1, t_2, \cdots, t_r\}$) the agents spend in arriving at these locations. We refer to the time elapsed as the *weight of an agent* at a given location. The solution presented in the previous section II-A turns out to be a special case of this problem with zero weights at all agent locations.



(a) Rendezvous at R (b) Rendezvous at TORP, R_t

Fig. 3. Illustration of rendezvous points for three agents with non zero weights

Fig. 3 illustrates the rendezvous of three agents located at P_1, P_2, P_3 with their associated weights t_1, t_2, t_3 respectively. It is worth noting that the information on locations of these agents prior to their arrival at P_1, P_2, P_3 is unknown and the figure illustrates only one among infinite possibilities of such locations. An arbitrary rendezvous location R is picked in Fig. 3(a). It can be observed that agents P_1 and P_2 arrive at R prior to the arrival of P_3. Thus the rendezvous time in this scenario is equivalent to the travel time of agent P_3 which is $T + t_3$.

In order to obtain the TORP, it is desirable to minimize the maximum time of arrival at a given rendezvous point. This is achieved by observing the weights at each agent location and ensuring that the rendezvous point be closer to the agent that has the highest weight. It is further desirable to verify the existence of a location where the agents can arrive simultaneously, as illustrated in Fig. 3(b). Such a location (if it exists) would have the following property given by (1). In the absence of weights, such a location would be the center of the smallest enclosing circle (Theorem 1). We utilize this to present Theorem 2.

$$t_1 + T_1 = t_2 + T_2 = t_3 + T_3 \tag{1}$$

Theorem 2: Consider r circles with their centers located at agent locations $P_1, P_2, \cdots, P_r$ and their radii equal to the weights at each location $t_1, t_2, \cdots, t_r$.

The TORP (R_t) for these r agents with their respective weights is the center of the smallest enclosing circle that contains each of these r circles.

Proof: Given the weight and agent location, the locus of points that can be reached from the agent location in fixed time constitutes a circle with center at the agent location and radius equal to its weight. This circle indicates all possible initial locations for an agent to arrive at P_i in time t_i $\forall i \in \{1, 2, \cdots, r\}$.

Given the initial locations of agents, TORP can be computed as the center of the smallest enclosing circle containing all the initial locations, as given by Theorem 1. Consequently, TORP for this problem can be computed as the center of the smallest enclosing circle that contains all the circles constructed at agent locations. **Q.E.D.**

We now present Lemma 1 which restricts the number of agent locations that constitute the smallest enclosing circle. However, the following mathematical facts are necessary in constructing the proof for lemma. For any two circles centered at C_1 and C_2 with radii r_1 and r_2, circle at C_1 is contained in circle at C_2 if (2) is satisfied. Additionally, the locus of points P where two agents located at P_1 and P_2 (with weights t_1 and t_2) arrive simultaneously turns out to be a hyperbola as given by (3), where c is a constant.

$$\overline{C_1C_2} \leq r_2 - r_1 \tag{2}$$
$$\overline{PP_1} + c \times t_1 = \overline{PP_2} + c \times t_2 \tag{3}$$

Lemma 1: The smallest enclosing circle C_t for r circles with non-zero radii requires a maximum of three circles for its construction. The remaining circles either lie in the interior of C_t or are tangential to its boundary.

Proof: It follows from (1) that the center of the smallest enclosing circle is located such that the associated agents arrive at it simultaneously. Such a center is the point of intersection of hyperbolas constructed with the locations of associated pairs of agents as given by (3). However, only a maximum of three hyperbolas can intersect at a single point in the plane (excluding degeneracy) which proves the first statement of the lemma.

Among all the points of intersections of hyperbolas, the point (say R_t) which maximizes the rendezvous time (T_t) is the TORP. Further, this rendezvous time is the radius of the smallest enclosing circle, centered at R_t. It thus follows from Theorem 2 that every other circle is contained (internal or tangential) in the circle C_t. **Q.E.D.**

We utilize Theorem 2 and Lemma 1 to present an algorithm that accepts the agent locations and weights as input and computes the TORP. We adopt an incremental approach by beginning with the largest circle, identifying the circles that lie external to it and gradually increasing its radius to enclose all the remaining circles.

Algorithm *Min_Time_Weights*

INPUT: Locations of all r agents $P_1, P_2, \cdots, P_r$ and their corresponding weights $t_1, t_2, \cdots, t_r$.

OUTPUT: Time Optimal Rendezvous Point (TORP), R_t and the time for rendezvous, T_t.

Step 1: Construct r circles with centers at agent locations and radii equal to their weights. Initialize an empty set S.

Step 2: Evaluate R_t and T_t as the location and weight of the agent with the highest weight and add its location to set S.

Step 3: For a circle C_t centered at R_t with a radius of T_t, use (2) to identify the agent whose circle is not contained in C_t. Add the agent location to S.

If all agents' circles are contained in C_t, output the current values of R_t and T_t. **Stop**.

Step 4: Evaluate R_t and T_t using **Step 5** on the set S and return to **Step 3**.

Step 5: If S has only two agent locations, evaluate R_t as the point of intersection of line segment joining them and the hyperbola constructed using (3).

If S has three agent locations, overwrite R_t with the point of intersection of three hyperbolas constructed using (3) and T_t with its corresponding weight.

If S has four agent locations, identify the triplet whose point of intersection of hyperbolas has the highest weight. Overwrite R_t with this point, T_t with its weight and remove the remaining agent location from S. ■

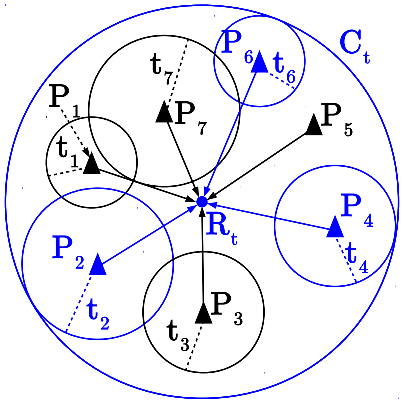


Fig. 4. TORP (R_t) for 7 agents with non-zero weights at their locations

Fig. 4 illustrates **Algorithm *Min_Time_Weights*** for 7 agent locations with their corresponding weights. The algorithm begins by constructing r circles as shown and picks agent P_2 that has the maximum weight. Step 3 of the algorithm identifies that the circle at P_3 lies outside the previous circle at P_2. Set S currently includes P_2 and P_3 which is used in computing and updating TORP in Step 5. Further iterations slowly increases the radius T_t of the smallest enclosing circle eventually computing C_t as shown. The final elements of the set S are P_2, P_4 and P_6 (shown in blue) which are used in constructing R_t in Step 5.

III. Algorithm to Compute TORP for Multi-Agent Systems Moving Amidst Obstacles

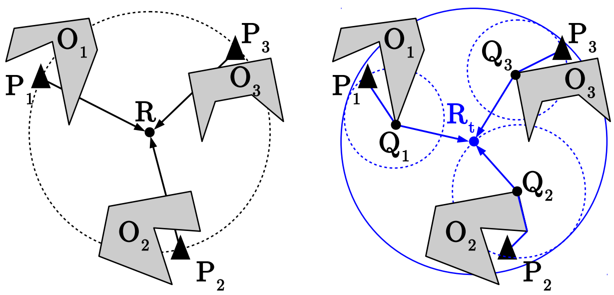
In this section, we compute the point that minimizes the total time for rendezvous for r agents as they negotiate m polygonal obstacles. In order to minimize the time for travel between two locations amidst obstacles, an agent computes and follows the shortest path from one location to another. This shortest path amidst m polygonal obstacles can be efficiently computed using the algorithm presented in [14]. We use this along with the results presented so far to develop the following Theorem 3 that computes the TORP.

Consider r agents denoted by $P_1, P_2, \cdots, P_r$ moving amidst m polygonal obstacles. Let R be the rendezvous point computed using **Algorithm *Min_Time_Weights*** with initial locations and zero weights. Let $Q_1, Q_2, \cdots, Q_r$ represent the agent locations before arriving at R when the agents take the shortest path from their initial locations (to R). Let $t_1, t_2, \cdots, t_r$ be the corresponding time taken by each agent.

*Theorem 3: TORP (R_t) for these r agents amidst m obstacles is the rendezvous point computed on the locations $Q_1, Q_2, \cdots, Q_r$ with their corresponding weights $t_1, t_2, \cdots, t_r$, using **Algorithm Min_Time_Weights**.*

Proof: The minimum time for rendezvous for r agents in the absence of obstacles occurs at R, as given by Theorem 1. Any deviation from the path to R increases the time for rendezvous. Since the deviation is minimum along the shortest path in the presence of obstacles, the locations given by $Q_1, Q_2, \cdots, Q_r$ are common to the paths taken by agents to arrive at both R and the TORP, R_t.

Further, the paths from agent locations $Q_1, Q_2, \cdots, Q_r$ to either R or R_t are not obstructed by any obstacle. It follows from Theorem 2 that the center of the smallest enclosing circle with corresponding weights would have a lower time for rendezvous than any other point in the plane, including R; which concludes that the center is indeed TORP (R_t). **Q.E.D.**



(a) Obstacles on the paths to R (b) TORP obtained using Theorem 3

Fig. 5. Computation of TORP for three agents amidst three obstacles

Fig. 5 illustrates rendezvous of three agents amidst three polygonal obstacles. The agents begin their attempt at rendezvous by computing the TORP using **Algorithm *Min_Time_Weights*** with zero weights at initial locations. The obstructions on the path to this rendezvous point R are shown in Fig. 5(a). In order to negotiate these obstacles, the agents have to deviate from their straight line path to R. Such a deviation necessitates a recomputation of TORP as indicated in Fig. 5(b).

The agents arrive at locations Q_1, Q_2 and Q_3 and recompute TORP using **Algorithm *Min_Time_Weights*** with weights equal to the time elapsed by the agents to arrive at these locations. The weights are represented by the radii of dashed circles. The new rendezvous point (R_t) that minimizes the time for rendezvous and the corresponding smallest enclosing circle are illustrated in Fig. 5(b). This is condensed into the following algorithm which is based on Theorem 3. An illustration for higher number of agents is presented via an experiment in section V.

Algorithm *Min_Time_Obstacles*

INPUT: Locations of all r agents and m polygonal obstacles. Weights at initial locations.

OUTPUT: Time Optimal Rendezvous Point (TORP), R_t and

the paths of all agents to R_t.
Step 1: Compute the rendezvous point R for r agents using their locations and corresponding weights with the help of **Algorithm *Min_Time_Weights***.
Step 2: Compute the shortest paths for all agents from their initial locations to R using [14]. Identify Q_i as the last vertex of obstacle visited by P_i before arriving at R along the shortest path, where $i \in \{1, 2, \cdots, r\}$.
If the shortest path for an agent P_i is not obstructed, identify Q_i as the agent location P_i.
Step 3: Compute TORP R_t using **Algorithm *Min_Time_Weights*** with locations $\{Q_1, Q_2, \cdots, Q_r\}$ and weights equal to the time taken by each agent to travel from P_i to Q_i. **Output** R_t and the corresponding shortest paths leading to R_t. ■

IV. Extension to Handle Dynamic Obstacles

In an industrial setting, the agents would have to negotiate dynamic obstacles like Automated Guided Vehicles (AGVs) or humans, en route to TORP. It is thus of interest to design an algorithm to ensure rendezvous, even when one or more agents are obstructed by dynamic obstacles. We begin by showing that subsequent re-computations do not affect the location of TORP when the agents negotiate just the static obstacles.

*Theorem 4: Recomputation of TORP (R_t) using **Algorithm Min_Time_Obstacles** at intermediate locations of agents after they have traveled for a finite time, does not affect its location.*
Proof: Time optimal rendezvous point is computed by finding the point of intersection of hyperbolas (given by (3)) at agent locations while taking their weights into consideration. This corresponds to Step 5 of **Algorithm *Min_Time_Weights***. Let the recomputation be performed at intermediate locations when every agent has traversed a finite time t_f. A constant factor of t_f thus appears in weights of agent locations as given by (4).

$$\overline{PP_1} + c \times (t_1 - t_f) = \overline{PP_2} + c \times (t_2 - t_f) \tag{4}$$

Since (4) evaluates to (3), there is no change in the point of intersection of hyperbolas and thus the TORP remains unaffected. **Q.E.D.**

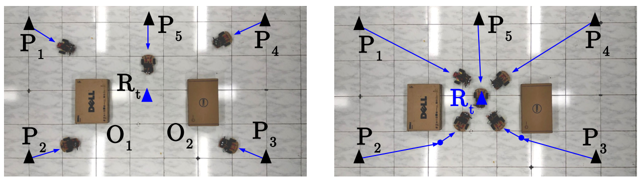
It is assumed that the agent, when obstructed by a dynamic obstacle, waits until its path is cleared. The time spent by an agent in waiting is not uniform across all agents. Thus the arguments in the proof to Theorem 4 no longer hold when even one of the agents faces a dynamic obstacle. It is thus necessary to recompute the TORP by taking into account the travel times of various agents and the time elapsed in waiting. This is accomplished with the help of the following algorithm.
Algorithm *Dynamic_Obstacle_Handling*
INPUT: Initial locations of all agents and static obstacles. Time Optimal Rendezvous Point (TORP), R_t and the paths of all agents to R_t. Distance sensor information.
OUTPUT: Rendezvous of all agents.
Step 1: Allow each agent to proceed on its path to R_t until faced by a dynamic obstacle or the agent arrives at R_t. If all agents arrive at the same location, **Stop**.
Step 2: If a dynamic obstacle is detected, halt the agent and communicate the current location of the agent along with the time elapsed in traveling and waiting, at fixed intervals. Request and receive this information from all other agents.
Step 3: Recompute and update the TORP (R_t) with the current locations of all agents and their corresponding weights (time elapsed), using **Algorithm *Min_Time_Obstacles***.
Step 4: Compute and update the shortest paths for all agents from their current locations to R_t (computed in **Step 3**). Proceed to **Step 1**. ■

V. Experimental Validation of Algorithms

The hardware realization of algorithms presented thus far is achieved with the help of small differential drive mobile robots. Each robot is equipped with an Arduino UNO board featuring an ATmega328P microcontroller to control the motion of robot and to compute the TORP with the location information on agents and obstacles. The communication between agents for exchanging information on location and elapsed time is achieved with the help of Xbee-PRO RF modules that operate at 2.4 GHz.

Detection of dynamic obstacles is achieved with the help of ultrasonic range detection sensors mounted on micro-servo motors. The localization of robots is attributed to MOC7811 speed sensor mounted on each wheel of the robot. MOC7811 is an inexpensive opto-coupler that provides adequate accuracy while eliminating any necessity for a motion capture system. A 12V, 1.3AH sealed maintenance-free lead acid battery is used to power each robot.

Various experiments have been performed to validate the proposed algorithms, two of which are presented here. Fig. 6 illustrates the first experiment where five agents are considered for rendezvous amidst two rectangular obstacles. Agents employ **Algorithm *Min_Time_Obstacles*** to compute the TORP, R_t. **Algorithm *Dynamic_Obstacle_Handling*** is then used by the agents to travel along their shortest paths to their destination R_t. Intermediate locations of agents are shown in Fig. 6(a) while their rendezvous is illustrated in Fig. 6(b).



(a) Agents proceed towards R_t (b) Rendezvous of agents at R_t

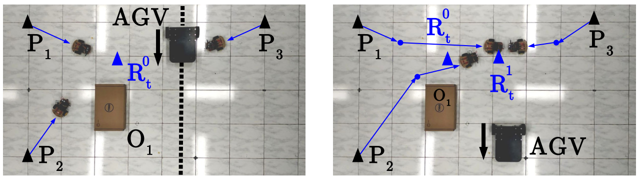
Fig. 6. Time-optimal rendezvous of five agents amidst two obstacles

In the second experiment (Fig. 7), we allow an AGV to obstruct the path of agent P_2 as three agents attempt to rendezvous at R_t^0 in the presence of one polygonal obstacle. The ultrasonic sensor on agent P_2 detects the AGV as a dynamic obstacle and invokes Step 2 of **Algorithm *Dynamic_Obstacle_Handling***. Once the current location information and the waiting times of all agents are communicated to each other, TORP is recomputed. While the agents P_1 and

TABLE I
COMPARISON OF VARIOUS FEATURES OF PROPOSED ALGORITHMS WITH PRIOR WORKS INVOLVING TIME-OPTIMAL RENDEZVOUS

Criteria $\rightarrow$	**Static and Dynamic Obstacles**	**Identical Agents**	**Hardware Realization**	**Complete Location Information**
[8], [11], [15]	No	No	No	Yes
[9], [10]	No	Yes	No	Yes
[16]	Yes	One	Yes	No
Proposed	Yes	Yes	Yes	Yes

P_3 keep moving to the current $TORP$, agent P_2 requests for recomputation until the AGV clears its path. It can be observed that the final rendezvous occurs at R_t^1.



(a) P_2 detects dynamic obstacle (b) Rendezvous at R_t^1

Fig. 7. Time-optimal rendezvous for three agents amidst one static and one dynamic obstacle (AGV)

Experimental results reflect the ability of proposed algorithms to quickly compute the TORP on just a microcontroller without any support from a central hub. The computation of shortest path from agents' locations to TORP is also parallelizable by allowing each agent to perform its own computation. Additionally, when handling a dynamic obstacle, the computation is performed in fixed intervals to further minimize the waiting times of agents.

VI. COMPARISONS

The algorithms presented in this paper have been compared with prior work in Table I. Prior work considering obstacles is, in general, limited. While there have been attempts to achieve time optimal rendezvous in dynamic cluttered environments [16], the study (and experiments) are limited to a single autonomous vehicle attempting a rendezvous with moving targets. In order to enhance our algorithm to handle dynamic obstacles, we allow the agents to communicate when faced by a dynamic obstacle and recompute the TORP by taking into account their waiting time, as discussed in section IV. Table II presents a comparison of the key features of experiments in our work and in [16].

VII. CONCLUSIONS

We have considered the rendezvous problem for multi-agent systems amidst obstacles in this paper. Various algorithms are presented for finding the Time Optimal Rendezvous Point (TORP) in the absence and presence of static obstacles. An algorithm to handle dynamic obstacles while retaining the constraint on minimum time is also presented. Finally, efficient hardware realization of the algorithms on indigenously fabricated small mobile robots is described.

TABLE II
COMPARISON OF VARIOUS ASPECTS OF EXPERIMENTAL SETUP

Criteria $\downarrow$	**[16]**	**Proposed**
Number of agents	2	r where $r \in \mathbb{Z}^+, r \geq 2$
Communication with Central Computer	Yes	No
Localization	CCD Camera	On-board Encoders
Processing Support	Central Host Computer	On-board Microcontroller

REFERENCES

[1] R. C. Luo and C. W. Kuo, "Intelligent seven-dof robot with dynamic obstacle avoidance and 3-d object recognition for industrial cyber-physical systems in manufacturing automation," *Proceedings of the IEEE*, vol. 104, no. 5, pp. 1102–1113, May 2016.

[2] M. S. Rahman, M. A. Mahmud, A. M. T. Oo, and H. R. Pota, "Multi-agent approach for enhancing security of protection schemes in cyber-physical energy systems," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 2, pp. 436–447, April 2017.

[3] B. Vogel-Heuser, C. Diedrich, D. Pantfrder, and P. Ghner, "Coupling heterogeneous production systems by a multi-agent based cyber-physical production system," in *2014 12th IEEE International Conference on Industrial Informatics (INDIN)*, July 2014, pp. 713–719.

[4] S. L. A. Cruz and B. Vogel-Heuser, "Comparison of agent oriented software methodologies to apply in cyber physical production systems," in *2017 IEEE 15th International Conference on Industrial Informatics (INDIN)*, July 2017, pp. 65–71.

[5] A. Ganguli, J. Cortés, and F. Bullo, "Multirobot rendezvous with visibility sensors in nonconvex environments," *IEEE Transactions on Robotics*, vol. 25, pp. 340–352, April 2009.

[6] J. Yu, S. LaValle, and D. Liberzon, "Rendezvous without coordinates," *IEEE Transactions on Automatic Control*, vol. 57, pp. 421–434, February 2012.

[7] J. Qin, Q. Ma, Y. Shi, and L. Wang, "Recent advances in consensus of multi-agent systems: A brief survey," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 6, pp. 4972–4983, June 2017.

[8] T. Setter and M. Egerstedt, "Minimum time power-aware rendezvous for multi-agent networks," in *2014 IEEE Conference on Control Applications (CCA)*, Oct 2014, pp. 2159–2164.

[9] G. Notarstefano and F. Bullo, "Distributed consensus on enclosing shapes and minimum time rendezvous," in *Proceedings of the 45th IEEE Conference on Decision and Control*, Dec 2006, pp. 4295–4300.

[10] A. Bhatia and E. Frazzoli, "Decentralized algorithm for minimum-time rendezvous of dubins vehicles," in *2008 American Control Conference*, June 2008, pp. 1343–1349.

[11] H. Chunhe and C. Zongji, "Minimum time rendezvous for multi-vehicle with non-identical velocity constraints," in *Proceedings of 2014 IEEE Chinese Guidance, Navigation and Control Conference*, Aug 2014, pp. 1718–1723.

[12] E. Wynters and J. Mitchell, "Shortest paths for a two robot rendezvous," in *Proc. of Fifth Canadian Conference on Computational Geometry*, 1993, pp. 216–221.

[13] B. Vundurthy and K. Sridharan, "Multi-agent gathering with collision avoidance and a minimax distance criterion - Efficient algorithms and hardware realization," *IEEE Transactions on Industrial Informatics*, To appear.

[14] J. Hershberger and S. Suri, "An optimal algorithm for euclidean shortest paths in the plane," *SIAM Journal on Computing*, vol. 28, no. 6, pp. 2215–2256, 1999.

[15] T. L. Brown, T. D. Aslam, and J. P. Schmiedeler, "Determination of minimum time rendezvous points for multiple robots via level set methods," in *ASME 2011 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, vol. 6, Aug 2011, pp. 787–797.

[16] F. Kunwar, F. Wong, R. B. Mrad, and B. Benhabib, "Time-optimal rendezvous with moving objects in dynamic cluttered environments using a guidance based technique," in *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Aug 2005, pp. 283–288.