Konya Mühendislik Bilimleri Dergisi., c. 9, s. 4, 965-980, 2021
Konya Journal of Engineering Sciences v. 9, n. 4, 965-980, 2021
ISSN: 2667-8055 (Elektronik)
DOI: 10.36306/konjes.833557

ADAPTIVE OPERATION MODEL FOR INTERIOR SMART LOGISTICS IN CYBER PHYSICAL SYSTEMS

[1] Emrah DÖNMEZ , [2] Fatih OKUMUŞ , [3]Adnan Fatih KOCAMAZ

[1]Bandırma Onyedi Eylül University, Department of Software Engineering, Balıkesir, TURKEY
[2]İnönü University, Department of Software Engineering, Malatya, TURKEY
[3]İnönü University, Department of Computer Engineering, Malatya, TURKEY
[1] emrahdonmez@bandirma.edu.tr, [2] fatih.okumus@inonu.edu.tr, [3] fatih.kocamaz@inonu.edu.tr

(Geliş/Received: 30.11.2020; Kabul/Accepted in Revised Form: 23.09.2021)

ABSTRACT: Logistics operations are among the main activities in industrial production areas. Today, vehicles that are usually electric and manually operated by a driver are used to perform these operations. Logistics robots are an important alternative that can be used in this field, and their use in integration with cyber physical systems in industrial fields is increasingly common. The biggest advantage of the logistics robots is that they provide autonomous driving capabilities and optimum parameters for the entire system in accordance with industry 4.0 concept. In this study, an adaptive logistics robot system that can be integrated into the Cyber Physical System (CPS) system in an environment with cyber physical system infrastructure has been developed. In this context, positioning, path planning, multi-task allocation, energy management, task prioritization, optimization and obstacle avoidance issues are analyzed and simple solutions are proposed. The experiments have been carried out in eight different configurations and the average distance and energy costs have been improved by 5.1% and 6.6%, respectively.

Key Words: *Cyber Physical System (CPS), Smart Logistics, Autonomous Robots, Industry 4.0, Robotic Control*

Siber Fiziksel Sistemlerde İç-Mekân Akıllı Lojistik için Adaptif İşletim Modeli

ÖZ: Lojistik operasyonlar, endüstriyel üretim alanlarında ana faaliyetler arasındadır. Günümüzde bu işlemleri gerçekleştirmek için genellikle elektrikli olan ve bir sürücü tarafından manuel olarak çalıştırılan araçlar kullanılmaktadır. Lojistik robotlar bu alanda kullanılabilecek önemli bir alternatiftir ve endüstriyel alanlarda siber fiziksel sistemlerle entegrasyonda kullanımları giderek yaygınlaşmaktadır. Lojistik robotların en büyük avantajı, endüstri 4.0 konseptine uygun olarak tüm sistem için otonom sürüş kabiliyetleri ve optimum parametreleri sağlamasıdır. Bu çalışmada, siber fiziksel sistem altyapısı olan bir ortamda Siber Fiziksel Sistem (SFS) içerisine entegre edilebilen uyarlanabilir bir lojistik robot sistemi geliştirilmiştir. Bu kapsamda konumlandırma, yol planlama, çoklu görev dağılımı, enerji yönetimi, görev önceliklendirme, optimizasyon ve engellerden kaçınma konuları analiz edilerek basit çözümler önerilmektedir. Deneyler sekiz farklı konfigürasyonda gerçekleştirilmiş ve ortalama mesafe ve enerji maliyetleri sırasıyla % 5,1 ve % 6,6 oranında iyileştirilmiştir.

Anahtar Kelimeler: *Siber Fiziksel Sistem (CPS), Akıllı Lojistik, Otonom Robotlar, Endüstri 4.0, Robotik Kontrol*

1. INTRODUCTION

Cyber-physical system (CPS) is the general name given to system in which physical mechanisms are controlled or monitored by computer-based algorithms. Cyber physical systems are more commonly used as industrial production technologies develop. These systems are one of the most significant components for the multi-agent based robotic control and monitoring systems. Because CPS is used to monitor an environment or control electro-mechanic systems (robots, production lines etc.) to optimize, stabilize and/or coordinate sub-components of the controllable environments. It is possible to adapt the cyber-physical system designed for a general production line to perform different tasks such as control, monitoring and intervention. The CPS systematically ensures fast and accurate interventions, coordinates physical systems, organize and transfer processed/measured data. This system can also provide efficient and low-cost solutions for multi-robot controlling and monitoring processes as well. One of the most common physical elements are mobile robots in the multi-robot management systems.

Mobile robots are interdisciplinary technological systems consisting of electronic, mechanical and software components that perform progress and maneuvering behaviors on a floor with moving transmission organs such as wheels, tracks, etc., and sensors for sensing the movement pattern / environment. They have been used for a wide variety of purposes in many sectors. Mobile robots become major actors in industrial production centers day by day. They collaborate collectively to perform given tasks in the environment. This collaboration is required multi-robot management and monitoring infrastructure to distribute and organize the available tasks. There are systems called smart and autonomous logistic systems where each agent collaborates one another to perform load handling tasks. All these management and monitoring processes are handled by the cyber-physical systems. Configuration of the CPS are adjusted according to the environment conditions and available tasks. The mobile robots are generally used to create smart logistics systems in industrial applications.

Smart logistics (SL) are implemented in industrial production areas, sea-ports, cargo management fields, cargo train ports and etc. Main advantages of such SL systems are that providing systematical load organization, smart scheduling, cooperation, safety, optimization (capacity, path and motion behaviors), cost reduction and so on. SL systems can also be managed and monitored by a CPS including computer-based algorithms. There are several issues in CPS-assisted SL systems. System integration (Iarovyi, Mohammed, Lobov, Ferrer, & Lastra, 2016) is a major difficulty to design an efficient CPS managing the elements of the SL system. The difficulty of system integration generally stems from interactions and conflicts between physical system, software and platform domains. The other concern about SL is safety in human environment. Even though researching robotics and facilitating technologies for applications are forming remarkable progress, it is important to create novel approaches for the design, modeling, and control of robotic systems operating safely with people hosted spaces (Schirner, Erdogmus, Chowdhury, & Padir, 2013). The SL systems have distributed system (Sztipanovits et al., 2012) in networked CPS elements. The distributed system provides required message transfer to create interaction between these networked elements. One of the most important component of the SL systems is Internet of Things (IoT) infrastructure (Wan et al., 2016). It simply provides accessibility to CPS objects regardless of location over the internet. Ultimately, by building smart data transfer layers between these components of SL systems, a stable coordination between CPS elements will have been done.

The organization of the paper is as follows: In the section 2, related works are given. The materials and methods are described in section 3. Experimental results and observations are emphasized in section 4. The conclusion and discussion are mentioned in section 5. The references are given in last section 6.

2. RELATED WORKS

There are remarkable number of studies related to CPS-structured SL systems. In each study carried out, a different module for CPS is considered according to the usage purpose and environment characteristics of multiple mobile robots. Main issues focused on these studies are; real-time communication, creation of optimized path plans, transportation safety, internet of things, sensor fusion, multi-robot management and coordination, etc.

A Platform-as-a-Service (PaaS) model (Krainer & Kirsch, 2014) has been introduced for implementing information acquisition missions from multi-client. Their system manage swarm of unmanned vehicles operated and maintained by an external infrastructure. They emphasize that the proposed system is integration of CPS and cloud computing as PaaS. A CPS system (Krueger et al., 2016) is proposed that it enables manufacturing tasks to be implemented with no intervention done by human. It is claimed that the proposed system operates even if the available environment definition (the world model) distresses from major uncertainties. By this way, the robot turns into an essential part of the manufacturing execution system. They said that the system has resulted concluding in a highly manageable complete system. In another study (Li et al., 2016), it has aimed to form a specific language called Cyber-Physical Robot Language (CPLR), that assists the cyber physical robot systems development and application in an efficient way. D'auria and Persia have highlighted the relevance of the application of CPS systems in the scope of robotic surgery (D'Auria & Persia, 2017). They introduce the design methodology for a cyber-physical system of collaborative robotics. This CPS model has been configured for surgery implementations aiming to minimize the vulnerability of robotic surgery systems. Semwal et al. propose a method to deal with the task execution order problem within a CPS that naturally organizes mutual exclusion (Semwal, Jha, & Nair, 2017). The method addresses a decentralized and distributed CPS that includes nodes such as computers, robots, and sensor nodes, and uses mobile software modules that help execute various tasks while also providing mutual exclusion of shared resources. Ernst has said that the robot autonomous vehicles (taxis) can decide where and when to go (Ernst, 2018). If necessary, it can also be controlled remotely, rather than autonomous driving, at least at the time of market entry where operating costs are less important. Besides, another important requirement is the ability to deal with faults. This requirement includes design and software faults as well as hardware faults and malfunctions. Laux et al.) have introduced the biological process required for sound signal based localization (Laux et al., 2018. In the study, a system that transfers single steps to a technical model is proposed to acquire a new type of localization model that uses one or two microphones to track an object. Their learning model is based on Support Vector Machines and Principal Component Analysis methods. Lee and Yang have investigated Massive MIMO (Multi-Input Multi-Output) usage infrastructure as a wireless technology in CPS (B. M. Lee & Yang, 2018). They have analyzed the overall performance of Massive MIMO base station positioned in a data center to ensure a dense connection to multiple devices for CPS based configurations. Lee et al have developed CPS-based intelligent robotic warehouse model to transform conventional warehouse tasks into smart logistics (C. K. M. Lee, Lin, Ng, Lv, & Tai, 2019). The multiple mobile robot resource allocation problem and calculation of collision-free path planning have been discussed in the study. An improved A* method has been designed to discover all possible routes between source node and target node. Levshun et al. have introduced the novel model, that is a key factor of the design and verification methodology for secure cyber-physical systems (Levshun, Chevalier, Kotenko, & Chechulin, 2020). While the developed model expresses cyber-physical systems as several building blocks, each building block is the projection of the integrated model with the properties and connections between them. Yaacoub et al. have given a general review to all components in CPS systems and their connectivity, including IoT systems (Yaacoub et al., 2020). The study describes the main CPS security threats, vulnerabilities and attacks related to the utilized components and communication protocols. Finally, current CPS security models that can be classified as cryptographic and non-cryptographic solutions are discussed and analyzed. Lu and Asghar have introduced the concept of semantic CPS (SCPS) and developed technologies which allow semantic communication between SCPSs (Lu & Asghar, 2020). The SCPS presents a layered architecture that separates the implementation of traditional CPS and semantic communication. This structure makes that it is compatible with various CPS application architectures. Dönmez and Kocamaz have proposed a load balancing and task distribution system for multi agents in their study (Dönmez & Kocamaz, 2019a; Dönmez & Kocamaz, 2019b). They have claimed that the proposed load balancing schemes generally provide an optimum balance between the number of tasks and path lengths for the multi agent systems. They proposed a path planning and smoothing method in an obstacle hosted environment (Dönmez, Kocamaz, & Dirik, 2017). In another study; (Dönmez & Kocamaz, 2019), a vision-based path planning system is developed by using multi-cameras for CPS based environments. Okumuş and Kocamaz proposed path planning methods (Okumus & Kocamaz, 2018) for multi- robots and developed a cloud platform (Okumuş & Kocamaz, 2019a; Okumus & Kocamaz, 2019b; Okumus, Donmez & Kocamaz 2020) for the multi-agent navigation management.

As it can be seen from the literature, there are various type of studies which focus different components of the CPSs. Most of these studies have focused several components of the CPS environments. On the other hand, these studies have been generally performed at theoretical level. In this study, a comprehensive adaptive model for CPS-assisted SL systems have been proposed. The model has several layers to deal with multi mobile robot agents while providing smooth task scheduling, coordination, managing, planning, controlling, safety and monitoring. We have compared our study with the featured studies () in the following Table 1.

Table 1. Comparison of the developed system in prominent related works.

	Chowdhury and Maier	**Afrin et al.**	**Turner et al.**	**Schillinger et al.**	**Our Work**
Environment Assets Definition	✓	✓	✓	✓	✓
AGV Charging	-	-	-	✓	✓
Mapping	✓	-	-	✓	✓
Multi-AGV	-	✓	✓	✓	✓
Multi-Task Allocation	✓	✓	✓	✓	✓
Simultaneous Task Execution	✓	✓	✓	✓	✓
Coordination Type	Centralized	Centralized	Decentralized	Centralized	Centralized
Monitoring	-	-	-	-	✓
Simulation	-	✓	✓	✓	✓
Path Planning	-	-	-	STAP	D* Lite
Multi-AGV Collaboration	-	✓	-	✓	✓
Collision Avoidance	-	-	-	✓	✓
Pseudocode	✓	✓	✓	✓	✓
Cost Optimization	✓	✓	✓	✓	✓
Position Tracking	-	-	-	-	✓

3. MATERIAL & METHODS

The CPS systems are mainly utilized in industrial production areas. There are a number of algorithms to monitor physical devices, machineries, production lines etc. Some algorithms are used to control some of these devices, robotic units, transfer vehicles etc. The CPS assisted SL systems are generalized for regular industrial production procedures within the scope of this study.

3.1. Position detection

There are a number of methods to determine the global position of the logistic robot(s) and workload(s) (targets). Encoder based methods are used to locate position of the mobile robot. However, the encoder data may give rise to error accumulation. Since moving parts of the robot can slip, skid because of the floor conditions. Tag based position detection is another method to locate mobile robots in the operating environment. Image processing methods through camera or special scanning devices are used to detect tags. Such methods require more complex mathematical processes and high computing power. In this study we propose a RFID- based position detection by using RFID tags and reader. We placed the tags according to the several predefined parameters over the floor where the mobile robot can make movement process. These predefined parameters are signals (S_i, S_n), active task (T_{act}), distance (D_n) between the tags ($T_n \in T_A$), distance error (E_n) reader antenna pitch and tag locations ($C_{x,y}$) and antenna distance to the floor. The basic steps of the position detection algorithm are given below. The parameter P_{Rn} is the position of the n_{th} robot. The following Algorithm 1 shows the process of position detection.

Algorithm 1 Pseudocode of the position algorithm

```
function PositionDetection()
    $S_i \epsilon A_i$ ← signal to all tags
    received signals ← $S_n \epsilon T_A$
    while $T_{act} = true$
        $D_n$ of $T_n \in T_A$ ← Compute
        $E_n$ ← Compute
        $C_{x,y}$ ← Refine $C_{x,y}$ of $T_n$
        $P_{Rn}$ w.r.t. $T_n$ ← Calculate
    return $P_{Rn}$
```

3.2. Multi-task allocation

Multi task allocation is one of the major challenges for the multi-robots in the CPS configured environments. An optimized task allocation can provide good balance between performance, cost and safety. In this study, we define several major parameters to perform multi-task allocation. These parameters are environment map which shows static obstacles (e.g. machineries, product lines), position of robots, position of available tasks, task list, definition of prioritized tasks, energy level of each robot, number of available robots (some may be charging, some may require fixing etc.) and P2P communication with CPS infrastructure. The operating scheme of the proposed adaptive operation model for the logistic robots in CPS based environment is demonstrated in Figure 1. In the multi task allocation algorithm, the energy levels of the robots are questioned after the positions of targets and robots are determined. Then, the optimum task assignment is carried out with other parameters including positions and energy levels.

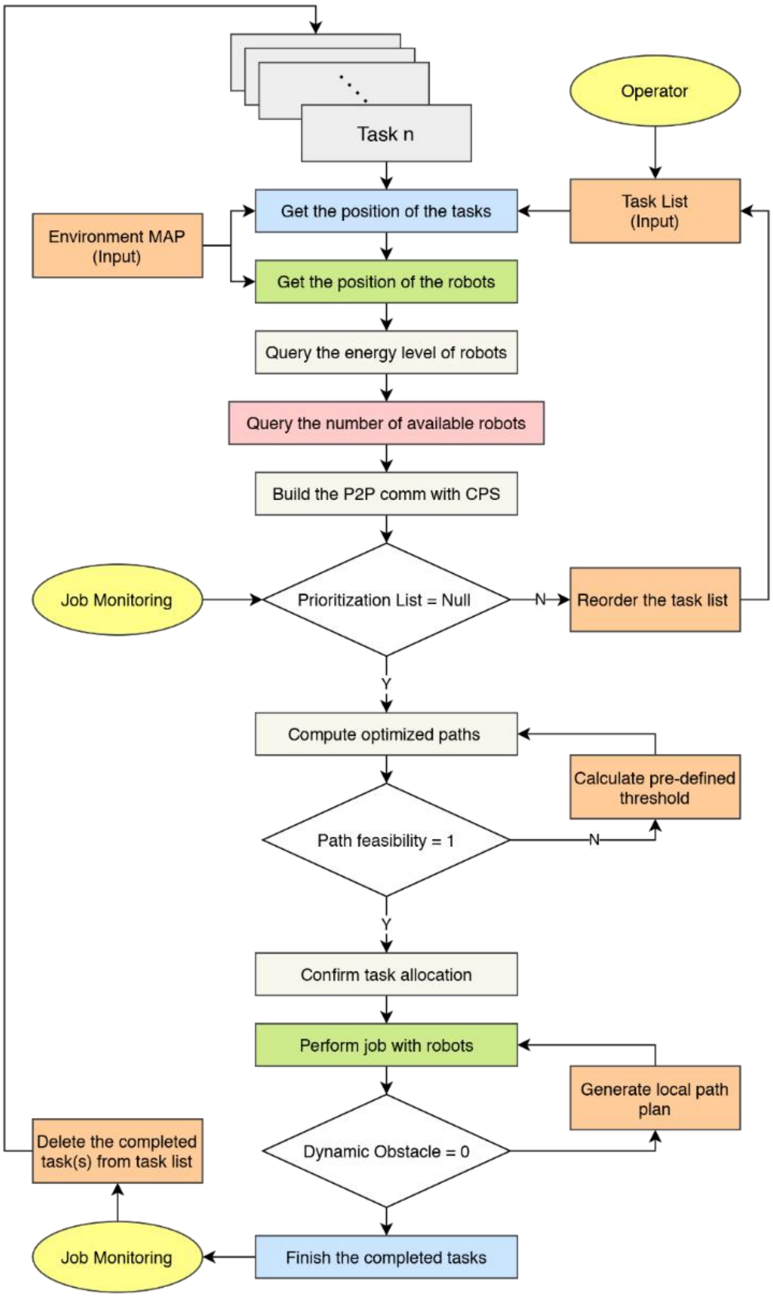
Task n
Operator
Task List (Input)
Environment MAP (Input)
Get the position of the tasks
Get the position of the robots
Query the energy level of robots
Query the number of available robots
Build the P2P comm with CPS
Job Monitoring
Prioritization List = Null
N
Reorder the task list
Y
Compute optimized paths
Calculate pre-defined threshold
Path feasibility = 1
N
Y
Confirm task allocation
Perform job with robots
Generate local path plan
Dynamic Obstacle = 0
Delete the completed task(s) from task list
Job Monitoring
Finish the completed tasks

Figure 1. The proposed adaptive operation model for logistic robots in CPS

3.2.1. Path planning

The path planning process is required to determine the tracking trajectory used by the logistic robots in the environment. There are multiple tasks, multiple robots and static/dynamic obstacles in the operating environment. There are three types of tasks in the environment; constant tasks, variable tasks and temporary tasks. The static obstacles are generally machineries etc. and the dynamic obstacles are generally other robots or humans etc. Therefore, path planning is performed by taking into account these variable/dynamic and constant/static parameters. There are a lot of path planning approaches in the literature. Dijkstra, A*, D*, APF, RRT, Probabilistic Roadmap etc. Each method has different advantages and complexities. The basic steps of the path planning algorithm are as following Algorithm 2. The ML_{TR} parameter the task-robot matching list, OP_n is optimal path, C_m is the conflicts, CPS is central management unit, W_s is warning signal and P_L is local path. The parameter R_n corresponds the

n_{th} robot. Similarly, the parameter O_{D_i} represents the i_{th} dynamic obstacle in the environment. T_n and R_n indicate the target (or workload) and robot, respectively.

Algorithm 2. Pseudocode of the path planning algorithm

```
function PathPlan()
    $ML_{TR}$ ← Target – Robot matching
    $OP_n$ ← between $T_n$ and $R_n$
    $C_m$ ← between $OP_n$
    if $C_m = true$ then
        recalculate $OP_n$
    if $C_m = false$ then
        $OP_n \rightarrow R_n$
    TrackRobot($R_n$)
    if $R_n$ send $O_{D_i} = true$ then
        run InformCentral() ← CPS management unit
    $R_n \rightarrow W_s$
    Determine $P_L$ to avoid $O_{D_i}$
    if $T_n$ is max then
    return $OP_n$ and $P_L$
```

3.2.2. Energy management

Logistic robots have battery pack to perform its motions. When the battery goes down to the critical level it needs to be recharged to maintain given tasks. If the tasks are distributed without considering the energy level of the robots, then there may be interruption between tasks and this interruption may affect the whole CPS environment. There may be production delay and critical processes increase the cost and decrease the efficiency. To overcome this problem, energy level of each robot (E_{L_n}) is checked and to perform the given T_n task energy consumption (EC_n) is calculated. If the energy level is not enough to complete the given task, then the R_n robot is directed to the charge station (CS_i). If the energy level of the robot decreases to the critical threshold level after completing the given task, then the robot is directed to the charge station to charge the battery, similarly. After battery analysis of the robot, the most suitable robot is assigned to the related task. The T_{list} and R_{next} represent the task list and the next available robot, respectively. The pseudocode of the energy management Algorithm 3 is given below.

Algorithm 3. Pseudocode of the energy management algorithm

```
function EnergyManagement()
    $EC_n$ of $R_n$ for $T_n \in T_{list}$ ← Calculate
    $E_{L_n}$ of $R_n$ for $T_n$ ← Check
    if $E_{L_n}$ is not feasible ↔ $EC_n$ then
        $R_n \rightarrow$ nearest $CS_i$
        $T_n \rightarrow$ nearest $R_{next}$
    return $E_{L_n}$
```

3.2.3. Task prioritization

The regular tasks for the logistic robots are realized according to the process order of robotic systems, machineries and conveyer belts etc. in the CPS. The order of processes is adjusted with respect to logical order of product processing. However, there may be a process that interrupts the whole logistic/production operations. Therefore, an approach is required to put in order the prioritized task/process in task list without stopping the operations. We have developed a task prioritization algorithm to optimize the system bay adding new tasks/jobs. The pseudocode of this process is given in Algorithm 4. The parameter T_{list} is the available task list. The Q_p prioritization queue holding prioritized tasks. The T_p represents the prioritized task. The L_{PT} label is used to mark T_p in T_{list}.

Algorithm 4. Pseudocode of the task prioritization algorithm

```
function Prioritization()
    $T_{list}$ ← Get
    $T_n \in T_{list}$ by FIFO ← Perform conventional queue balance
    $Q_p$ queue ← Check
    if $\forall\ T_p \in Q_p$ then
        Add $T_p$ to $T_{list}$ with $L_{PT}$ label
    $T_p$ acc. to $L_{PT}$ order in $T_{list}$ ← Perform
    completed $T_p \in T_{list}$ ← Remove
return
```

3.2.4. Path optimization

Path optimization is a critical issue in terms of cost reduction and efficiency increase. When the tasks are distributed among the available logistic robots in the CPS; firstly, positions of both robots and targets are identified. Then energy level of each robot is checked and shortest path between target and robots are calculated. The robot closest to the target is assigned to this target. This process is repeated for all available robots for logistics. If there are some targets (T_i) that are not assigned to any robot, those targets will be queued to be assigned to the appropriate robot that completed the given task. After robot-task assignment process, the possible conflicts (PC_{list}) between the paths are analyzed and alternative paths are created if there is a critical conflict that have to be eliminated. Another issue is prioritized tasks (Q_p) that defined after all general process are completed while robots perform the given tasks. These tasks may give rise to the changes in optimized path trajectories. The paths are recalculated if the prioritized tasks interrupt the logistic operation. The Algorithm 5 being pseudocode is summarized all these processes. The T_{list} and R_{list} are the available task and robot lists. The parameters P_s, T_m and R_n are path, task and robot, respectively. The P_{O_n} parameter represents the optimized n_{th} path between R_n and T_m.

Algorithm 5. Pseudocode of the path optimization algorithm

```
**function** PathOptimization()
    $T_{list}$ and $R_{list}$ ← Get
    $P_s$ between $T_m$ and $R_n$ ← Calculate
    $R_n \rightarrow T_m$
    store $P_{O_n}$ ← PathPlan()
    **if** $\forall\ T_i \in T_{list}$ that is not assigned to any $R_n$ **then**
        Queue $T_i \rightarrow Q_p$
    Calculate $PC_{list}$
    $PC_j \in PC_{list}$ ← Optimize conflicted path
    **if** $\forall\ T_p \in Q_p$**then**
        **return** $P_{O_n}$
```

3.2.5. Obstacle avoidance

There are two types of obstacles in the operating environment of CPS. The first type of obstacle is static obstacles and second type is dynamic obstacles. The static obstacles are again two types; permanent static obstacles and temporary static obstacles. The permanent static obstacles are machineries, conveyor belts, fixed electronic devices etc. The temporary static obstacles are obstacles which is placed to a position for a temporary time. The environment map is used to create the path plans. This map includes static obstacles. The permanent ones are predefined to the map. The temporary ones are defined to the map together a time label. Both obstacle types don't affect the path trajectories when robot have performed the assigned task(s). However, the main issue in terms of the obstacles are dynamic obstacles. Their behavior is not predictable to create a static path plan. When they emerge and which direction, they came are ambiguous. The dynamic obstacles can be detected with local (onboard) sensors (S_v) placed on the robots to overcome behaviors of dynamic obstacles. After detection process of the dynamic obstacles, the robot sends sound and light warning without attempting any movements. If the obstacle(s) does not respond, the operator(s) of the CPS infrastructure are informed. The robot creates the local path plan to avoid obstacle. This local path plan provides that the robot temporarily goes out from the main path plan and when the obstacle is passed the robot goes into the main path plan. In addition, the robot passes the obstacle with a predefined safe distance threshold. The dynamic obstacle avoidance Algorithm 6 is given below. The parameter O_{D_i} represents the i_{th} dynamic obstacle in the environment. The W_s is the warning signal mentioned before. The $OP_{(CPS)}$ corresponds to the CPS operator. The parameters P_L and P_M are local path and main path, respectively. The T_i is the i_{th} task actively processed.

Algorithm 6. Pseudocode of the obstacle avoidance algorithm

```
function ObstacleAvodiance()
    $S_v \leftarrow$ Get
    if $S_v\ != 0$ then
        $O_{D_i} = true$
        $W_s \rightarrow OP_{(CPS)}$
    if $O_{D_i} \rightarrow$ not respond then
        Create $P_L$ & Track $P_L$
    if $P_L$ completed $= true$ then
        Track $P_M$ until $T_i$ is completed
```

3.2.6. Job monitoring

The job (task) monitoring module performs the tasks of viewing the list of assigned tasks, monitoring the status of the current tasks and monitoring the completed tasks. In this context, it monitors the prioritized tasks and removes the completed tasks from the task list. In this way, the CPS makes the minimum number of operations to focus on the remaining tasks.

4. EXPERIMENTAL RESULTS AND OBSERVATIONS

There are four robots and four seven tasks (targets) for the experiments. Static obstacles are demonstrated with brown color. Each robot is a dynamic obstacle for the other robots. Humans other movable devices are assumed as dynamic obstacles as well. The experimental environment for the proposed adaptive operation model is demonstrated in the following Figure 2. Simulation environment is simply modeled in the MATLAB and Python programming environment. In the figure, R1, R2, R3 and R4 represent the robots. T1, T2, …, T7 correspond to the targets. The Rp1, Rp2, …, Rp5 are the representative RFID tags which identifies the key-positions in the main path planning.

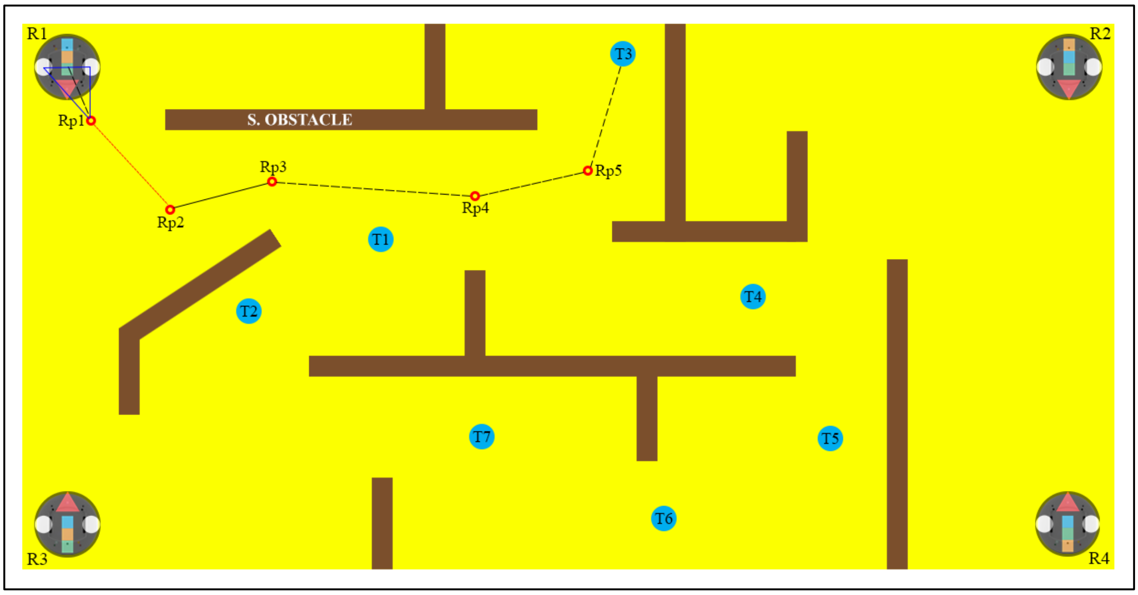

Figure 2. Representative operating environment

The RFID tags are used to locate the position of the robots. A sample path plan between the R1 robot and T3 target is shown in the figure as well. The number of robots and targets can be increased. However, to show details of the study with simple structures, a limited number of robots and targets are placed to the CPS system. Robots are simply identified onboard colored tags. Each colored tag has different order of colors to distinct robots. The tasks are distributed by taking into account both distance value and number of the tasks. The path plans are created for four robots between the seven tasks in an obstacle hosted environment. This path plans are demonstrated in Figure 3. The following Table 2 shows the local coordinates of the robots.

Table 2. Local (px) coordinates of the robots

Coord.	R1	R2	R3	R4
X	69	1170	69	1170
Y	67	67	571	571

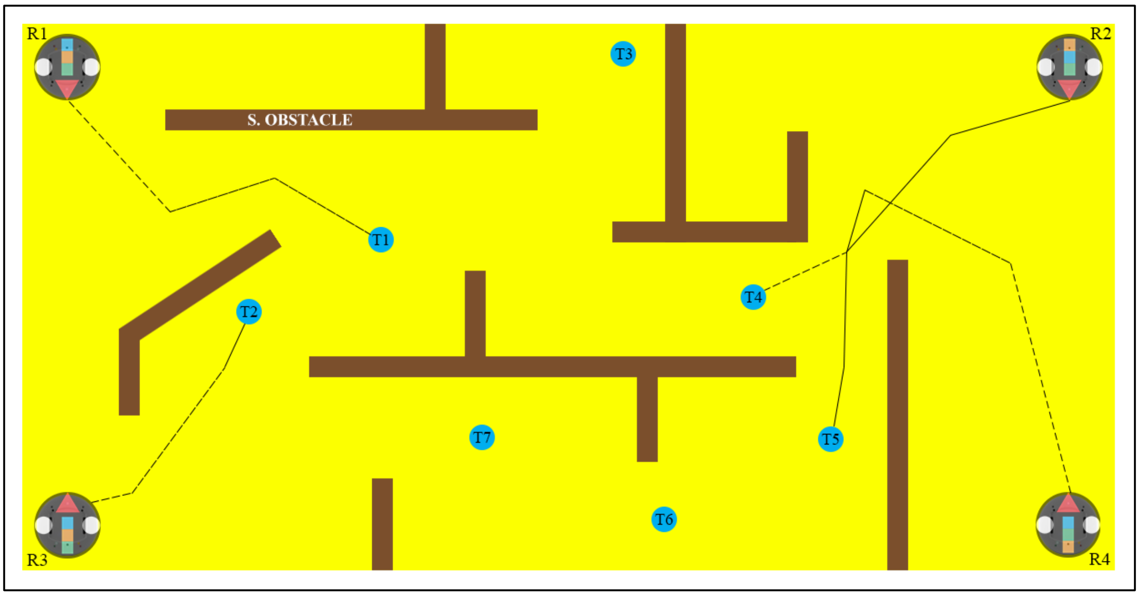

Figure 3. Assigned tasks to multiple robots and path plans

The multitask allocation algorithm detects the closest task/target to the robot and assign that target to the robot. The robots perform the assigned tasks simultaneously. When the two or more robot encounter in the operating environment while following the paths, the foremost task in the queue is prioritized. Therefore, the encountered other robots give the path to the robot assigned to this foremost task. These robots simply get out of the path by determining a suitable position. The following Figure 4 shows task allocation process when there are no obstacles in the environment.

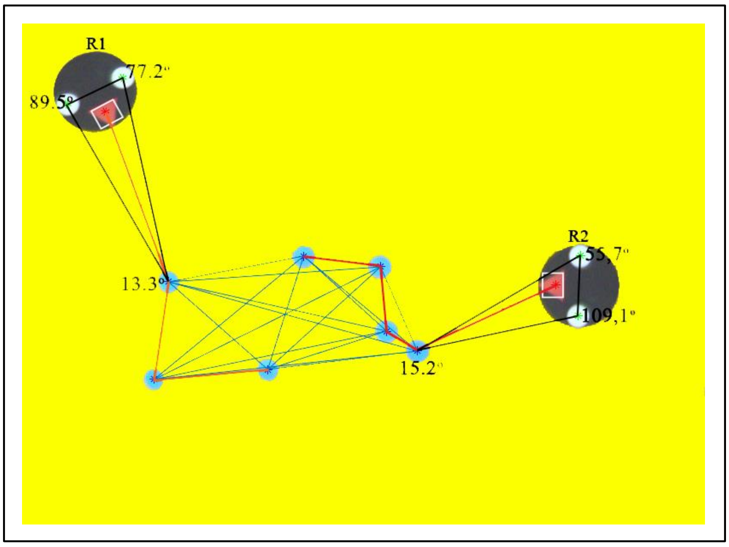

Figure 4. Multi Task allocation without any obstacle(s)

There are 7 task (or targets) and two robots in the Figure 4. The algorithm firstly calculates the distances between the robot and targets. Because of there is no target, distances between the targets are calculated as well. These process is made for all targets and a fully connected graph like structure is acquired. This structure is utilized to determine the shortest path between the allocated targets to the robots. The figure 5 demonstrates a prioritization process. The prioritized task T7 is shown with red color.

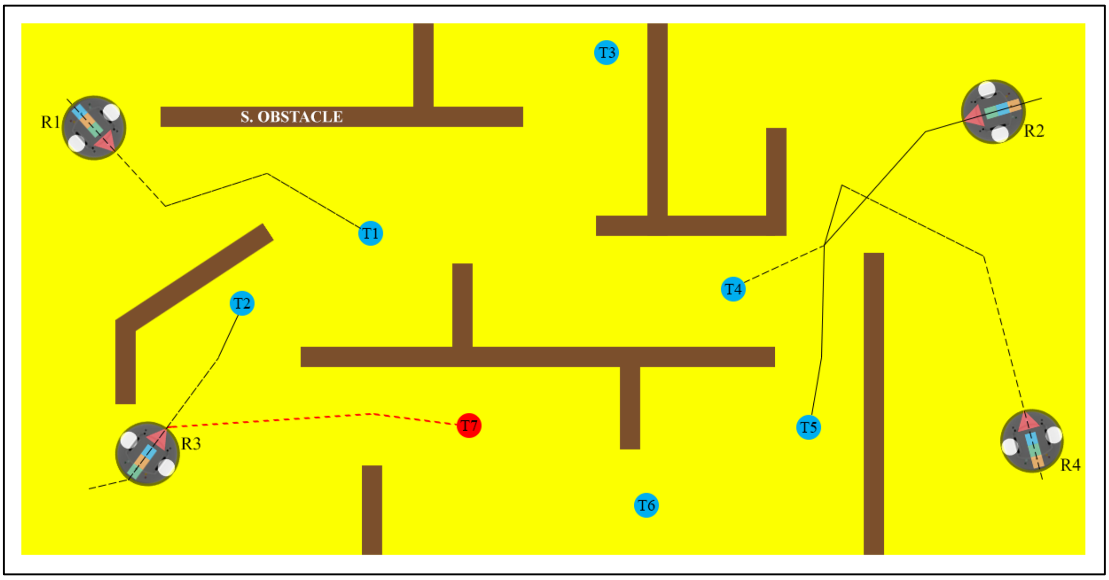

Figure 5. A task prioritization scheme

When a prioritized task is added to the task list, all the robots attempting to perform the assigned tasks are alerted about the prioritized task(s). The robots send the coordinate information of the current location to the CPS. Then the nearest robot to the prioritized task is directed to this target. The canceled task is added to the queue by checking the distances and available robot situations. Therefore, the path optimization algorithm is executed to check if there are any changes in the cost of the paths. The following Table 3 shows the local coordinates for robots when prioritization is employed.

Table 3. Local coordinates (px) of the robots

Coord.	R1	R2	R3	R4
X	102	1116	162	1160
Y	139	120	505	492

The Figure 6 shows the obstacle avoidance process between the robot and dynamic obstacle (another robot).

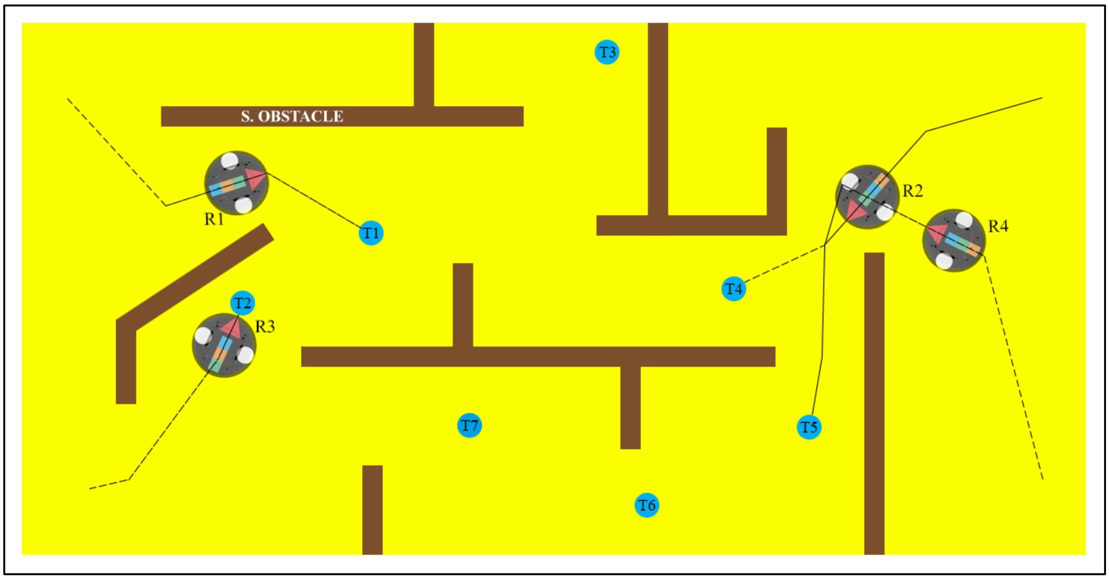

Figure 6. Dynamic obstacle avoidance process

The R2 and R4 robots have encountered when they trying to perform the assigned tasks. The R2 robot is a dynamic obstacle for the R4 robot in this situation. Therefore, the R4 robot waits for the R2 robot to pass toward to the T4 target. When the obstacle detection sensors of R4 have not detect any obstacles, the R4 robot maintains its movement to the assigned T5 target. The following Table 4 shows the local coordinates for robots when dynamic obstacle avoidance process is performed.

Table 4. Local coordinates (px) of the robots

Coord.	R1	R2	R3	R4
X	262	974	248	1070
Y	200	215	384	265

Several experiments have been conducted and their results are given in the following Table 5. The distance values are computed by utilizing Euclidean distance calculation.

Table 5. Experimental results for distance cost and energy

Experiments	Total Distance (px)	Optimized Total Distance (px)	Total Energy (kW)	Optimized Total Energy (kW)
Exp-1	1580	1492	21,6	20,1
Exp-2	1635	1523	23,7	21,4
Exp-3	1442	1411	20,4	19,8
Exp-4	1723	1608	25,8	23,3
Exp-5	1336	1288	19,1	18,6
Exp-6	1434	1368	20,3	19,6
Exp-7	1682	1603	23,9	23,3
Exp-8	1501	1414	21,3	19,4
Average	1541,63	1463,38	22,01	20,69

Seven targets and four robots have been used in the experiments. There are eight experiments and each experiment has different task/target distribution. Each experiment is repeated five times and average value of these experiments have been calculated and placed to the table. Total distance (without system optimization), optimized total distance, total energy (without energy optimization) and optimized total energy parameters have been calculated as the performance metrics to observe the overall system performance. The following equation 1 is utilized the path cost.

$$D_S = \sqrt{\sum_{i=1}^{k}(q_i - p_i)^2} \qquad (1)$$

In this equation, the D_S represents the distance of the simulated path. The k parameter (q_i, p_i) is the number of Cartesian coordinates. Parameter k is Cartesian coordinates given in the form of $(q_i\ , p_i)$ on the path. The cost of the path is made by combining n part consecutive edges between cartesian coordinates.

Total distances have been optimized according to the path optimization algorithm given before. Additionally, total energy has been calculated by a rate between the total and optimized distances without considering all the environmental factors (e.g. coefficient of friction). However, there may be several dynamic obstacle avoidance operations that increase energy consumption. Therefore, it may be necessary to ignore external factors such as obstacle avoidance. The average has been given to demonstrate the average metric values of the different task configurations. The distance and required energy values changes according to the distribution and position of the tasks. Both distance and energy have been improved with the proposed system by decreasing the total distances and total energy. The distance and energy gains changes in each experiment; the most gained value has been about 6,9% for distance and 9,8% for energy in second experiment (Exp-2). The average gains have been 5,04% and 5,99% for distance and energy optimization, respectively.

5. CONCLUSION AND DISCUSSION

The logistic robots in the CPS infrastructures are one of the key components for the overall automation performance. The main factors affecting the autonomous logistics hosted CPS environments are positioning, path planning, multi-task allocation, energy management, task prioritization, optimization and obstacle avoidance issues. All these issues are covered with simple and efficient approaches within the scope of this study. Specific models and pseudocodes of the proposed algorithms are given with their details. A simple simulation environment is created and experiments are conducted in this environment. The conducted experiments have showed that the proposed system can manage the different task configurations while improving the overall performance of logistics robot system being an important component of the CPS. In future, experiment will be conducted in a real laboratory environment. We plan to adapt deep learning and machine vision methods in this environment to strengthen the proposed models for the logistic robot automation.

6. ACKNOWLEDGEMENT

This study was carried out within the scope of the project with the FKP-2020-1993 project code of the Inönü University BAP unit. The study has been supported by Çalık Denim company and the company's R&D Center. The stages required for cyber physical systems have been determined in the company's production site. We would like to thank all institutions and centers that contributed to the study.

REFERENCES

Afrin, M.; Jin, J.; Rahman, A.; Tian, Y.C.; Kulkarni, 2019, "A. Multi-objective resource allocation for Edge Cloud based robotic workflow in smart factory.", *Future Gener. Comput. Syst.* 2019, 97, 119–130, doi:10.1016/j.future.2019.02.062.

Chowdhury, M.; Maier, M., 2017, "Collaborative Computing for Advanced Tactile Internet Human-to-Robot (H2R) Communications in Integrated FiWi Multirobot Infrastructures.", *IEEE Internet Things J.*, 4, 2142–2158, doi:10.1109/JIOT.2017.2761599.

D'Auria, D., & Persia, F., 2017, "A collaborative robotic cyber physical system for surgery applications.", In *Proceedings - 2017 IEEE International Conference on Information Reuse and Integration, IRI 2017*. https://doi.org/10.1109/IRI.2017.84

Donmez, E., & Kocamaz, A. F., 2019, "The eye-out-device multi-camera expansion for mobile robot control.", *2019 International Conference on Artificial Intelligence and Data Processing Symposium, IDAP 2019*. https://doi.org/10.1109/IDAP.2019.8875981

Dönmez, E., & Kocamaz, A. F., 2019, "Multi Target Task Distribution and Path Planning for Multi-Agents.", *2018 International Conference on Artificial Intelligence and Data Processing, IDAP 2018*. https://doi.org/10.1109/IDAP.2018.8620932

Dönmez, E., & Kocamaz, A. F., 2019, "Çoklu Hedeflerin Çoklu Robotlara Paylaştırılması İçin Bir Yük Dengeleme Sistemi." *Bitlis Eren Üniversitesi Fen Bilimleri Dergisi*. https://doi.org/10.17798/bitlisfen.467757

Dönmez, E., Kocamaz, A. F., & Dirik, M., 2017, "Bi-RRT path extraction and curve fitting smooth with visual based configuration space mapping.", *IDAP 2017 - International Artificial Intelligence and Data Processing Symposium*. https://doi.org/10.1109/IDAP.2017.8090214

Ernst, R., 2018, "Automated Driving: The Cyber-Physical Perspective", *Computer*. https://doi.org/10.1109/MC.2018.3620974

Iarovyi, S., Mohammed, W. M., Lobov, A., Ferrer, B. R., & Lastra, J. L. M., 2016, "Cyber-Physical Systems for Open-Knowledge-Driven Manufacturing Execution Systems.", *Proceedings of the IEEE*. https://doi.org/10.1109/JPROC.2015.2509498

Krainer, C., & Kirsch, C. M., 2014, "Cyber-physical cloud computing implemented as PaaS.", *Proceedings of the 4th ACM Workshop on Design, Modeling and Evaluation of Cyber Physical Systems, CyPhy 2014*. https://doi.org/10.1145/2593458.2593461

Krueger, V., Chazoule, A., Crosby, M., Lasnier, A., Pedersen, M. R., Rovida, F., … Veiga, G., 2016, "A vertical and cyber-physical integration of cognitive robots in manufacturing.", *Proceedings of the IEEE*. https://doi.org/10.1109/JPROC.2016.2521731

Laux, H., Bytyn, A., Ascheid, G., Schmeink, A., Kurt, G. K., & Dartmann, G., 2018, "Learning-based indoor localization for industrial applications.", *2018 ACM International Conference on Computing Frontiers, CF 2018 - Proceedings*. https://doi.org/10.1145/3203217.3203227

Lee, B. M., & Yang, H., 2018, "Massive MIMO for Industrial Internet of Things in Cyber-Physical Systems.", *IEEE Transactions on Industrial Informatics*. https://doi.org/10.1109/TII.2017.2787988

Lee, C. K. M., Lin, B., Ng, K. K. H., Lv, Y., & Tai, W. C., 2019, "Smart robotic mobile fulfillment system with dynamic conflict-free strategies considering cyber-physical integration.", *Advanced*

Engineering Informatics. https://doi.org/10.1016/j.aei.2019.100998

Levshun, D., Chevalier, Y., Kotenko, I., & Chechulin, A., 2020, "Design and verification of a mobile robot based on the integrated model of cyber-Physical systems.", *Simulation Modelling Practice and Theory*, *105*, 102151. https://doi.org/10.1016/j.simpat.2020.102151

Li, F., Wan, J., Zhang, P., Li, D., Zhang, D., & Zhou, K., 2016, "Usage-specific semantic integration for cyber-physical robot systems.", *ACM Transactions on Embedded Computing Systems*. https://doi.org/10.1145/2873057

Lu, Y., & Asghar, M. R., 2020, "Semantic communications between distributed cyber-physical systems towards collaborative automation for smart manufacturing.", *Journal of Manufacturing Systems*. https://doi.org/10.1016/j.jmsy.2020.05.001

Okumuş, F., & Fatih, A., 2019, "Exploring the Feasibility of a Multifunctional Software Platform for Cloud Robotics.", *2018 International Conference on Artificial Intelligence and Data Processing, IDAP 2018*. https://doi.org/10.1109/IDAP.2018.8620865

Okumus, F., & Kocamaz, A. F., 2018, "Comparing Path Planning Algorithms for Multiple Mobile Robots.", *2018 International Conference on Artificial Intelligence and Data Processing (IDAP)* (pp. 1–4). IEEE. https://doi.org/10.1109/IDAP.2018.8620785

Okumus, F., & Kocamaz, A. F., 2019, "Cloud based indoor navigation for ros-enabled automated guided vehicles.", *2019 International Conference on Artificial Intelligence and Data Processing Symposium, IDAP 2019*. https://doi.org/10.1109/IDAP.2019.8875993

Okumus, F., Donmez, E., & Kocamaz, A. F., 2020, "A Cloudware Architecture for Collaboration of Multiple AGVs in Indoor Logistics: Case Study in Fabric Manufacturing Enterprises.", *Electronics.* 9(12). https://doi.org/10.3390/electronics9122023

Schillinger, P.; Bürger, M.; Dimarogonas, D.V., 2018, "Simultaneous task allocation and planning for temporal logic goals in heterogeneous multi-robot systems.", *Int. J. Robot. Res.* 2018, 37, 818–838, doi:10.1177/0278364918774135.

Schirner, G., Erdogmus, D., Chowdhury, K., & Padir, T., 2013, "The future of human-in-the-loop cyber-physical systems.", *Computer*. https://doi.org/10.1109/MC.2013.31

Semwal, T., Jha, S. S., & Nair, S. B., 2017, "On ordering multi-robot task executions within a cyber physical system.", *ACM Transactions on Autonomous and Adaptive Systems*. https://doi.org/10.1145/3124677

Sztipanovits, J., Koutsoukos, X., Karsai, G., Kottenstette, N., Antsaklis, P., Gupta, V., … Wang, S., 2012, "Toward a science of cyber-physical system integration.", *Proceedings of the IEEE*. https://doi.org/10.1109/JPROC.2011.2161529

Turner, J.; Meng, Q.; Schaefer, G.; Whitbrook, A.; Soltoggio, A., 2017, "Distributed Task Rescheduling with Time Constraints for the Optimization of Total Task Allocations in a Multirobot System.", *IEEE Trans. Cybern.* 2018, 48, 2583–2597, doi:10.1109/TCYB.2017.2743164.

Wan, J., Tang, S., Shu, Z., Li, D., Wang, S., Imran, M., & Vasilakos, A. V., 2016, "Software-Defined Industrial Internet of Things in the Context of Industry 4.0.", *IEEE Sensors Journal*. https://doi.org/10.1109/JSEN.2016.2565621

Yaacoub, J. P. A., Salman, O., Noura, H. N., Kaaniche, N., Chehab, A., & Malli, M., 2020, "Cyber-physical systems security: Limitations, issues and future trends.", *Microprocessors and Microsystems*. https://doi.org/10.1016/j.micpro.2020.103201