Efficient Cooperation of Heterogeneous Robotic Agents

A Decentralized Framework

©SHUTTERSTOCK.COM/ZENZEN

In social and industrial facilities of the future such as hospitals, hotels, and warehouses, teams of robots will be deployed to assist humans in accomplishing everyday tasks like object handling, transportation, or pickup and delivery operations. In such a context, different robots (e.g., mobile platforms, static manipulators, or mobile manipulators) with different actuation, manipulation, and perception capabilities must be coordinated to achieve various complex tasks (e.g., cooperative parts assembly in the automotive industry or loading and unloading of palettes in warehouses) that require collaborative actions with each other and with human operators (Figure 1).

By Michalis Logothetis, George C. Karras, Konstantinos Alevizos, Christos K. Verginis, Pedro Roque, Konstantinos Roditakis, Alexandros Makris, Sergio García, Philipp Schillinger, Alessandro Di Fava, Patrizio Pelliccione, Antonis Argyros, Kostas J. Kyriakopoulos, and Dimos V. Dimarogonas

Digital Object Identifier 10.1109/MRA.2021.3064761
Date of current version: 23 April 2021

1070-9932/21©2021IEEE

The efficient supervision and coordination of a heterogeneous system mandates a decentralized framework that integrates high-level task planning, low-level motion planning and control, and robust real-time sensing of the robot's dynamic environment. Decentralization in multiagent robotic systems is of utmost importance because it provides flexibility, scalability, and fault-tolerance capabilities. In this work, we present the architecture of the decentralized framework developed within the context of the European Union project Co4Robots and its application in a multitasking collaboration scenario involving various heterogeneous robots and humans.

Common Applications

Multiple robots are commonly used in cooperative applications, such as exploration, surveillance, service robotics, and cognitive factories [1]. However, dealing with the collaboration of heterogeneous multirobot systems is a rather tricky undertaking due to the different kinematic and sensing capabilities of each robot. One issue of utmost importance in coordination of robotic teams is multiagent task planning and control. Significant efforts have been devoted toward this in the past decades, resulting in a number of high-complexity algorithms [2].

A standard classification that arises in multiagent planning and control is centralized versus decentralized schemes, depending on whether the assignment of actions to the agents is performed by a central computer unit or locally by each agent. Current practice in coordination of robotic teams is based on offline centralized planning, and related tasks are almost exclusively fulfilled in a predefined manner, allowing little room for real-time and coordinated decentralized actions. Centralized planning schemes with global and local tasks usually provided satisfactory results [3]; however, they have been proven to be computationally expensive. On the other hand, decentralized planning significantly reduces computational complexity [4]. A rather important issue in heterogeneous decentralized task planning is role assignment and task allocation, where each agent's capabilities diverge depending on its teammate and/or their mutual state [5].

From the control point of view, multirobot cooperative object manipulation and transportation have been well studied in the literature, especially in a centralized framework [6]. Despite its performance, centralized control is less robust since all units rely on a central system, and its complexity increases rapidly as the number of participating robots increases. On the other hand, decentralized control approaches usually depend on heavy, explicit interrobot communication and global offline knowledge of the desired task [7]. Nevertheless, in such tasks, implicit interrobot communication arises naturally as a side effect of the robot's physical interactions (e.g., the interaction forces between the object and the robot), which can be easily acquired by appropriate sensors attached to the robot [8]. However, limited studies have been conducted in cooperative object manipulation and transportation via heterogeneous robotic systems [9].

Hence, our work is motivated by the need to have multirobot decentralized systems, where a significant amount of information can be implicitly acquired via physical interactions and processed locally, reducing the need for exhaustive, explicit interrobot communication. More specifically, we present a complete decentralized framework consisting of 1) a set of perceptual algorithms that enable cooperating robots to estimate the state of their highly dynamic environment, 2) a set of control schemes appropriate for the mobility and manipulation capabilities of the considered robotic platforms, 3) a systematic real-time decentralized methodology to accomplish complex mission specifications given to a team of heterogeneous robots, and 4) the corresponding systematic integration of these modalities at both conceptual and software implementation levels. The efficacy of the overall framework is demonstrated via a complex scenario, which involves three heterogeneous robots and humans cooperating in loading and transportation tasks.

Decentralization in multiagent robotic systems is of utmost importance because it provides flexibility, scalability, and fault-tolerance capabilities.

System Components

Our purpose is to develop a decentralized framework that will be able to support logistic tasks in an automated manner by efficiently allocating a set of heterogeneous robotic agents as well as humans that collaborate appropriately according to the specifications of each task. Hence, we envision the employment of 1) a dexterous seven-degrees-of-freedom (DoF) static manipulator able to perform loading and unloading actions of light and heavy objects, 2) a mobile manipulator to extend the motion flexibility and reachability of the overall framework, and 3) a mobile robot for the transportation of objects across different areas of the workspace, endowed with the ability to ease the loading and unloading procedures by

Figure 1. Heterogeneous agents cooperating in a multitasking collaboration scenario.

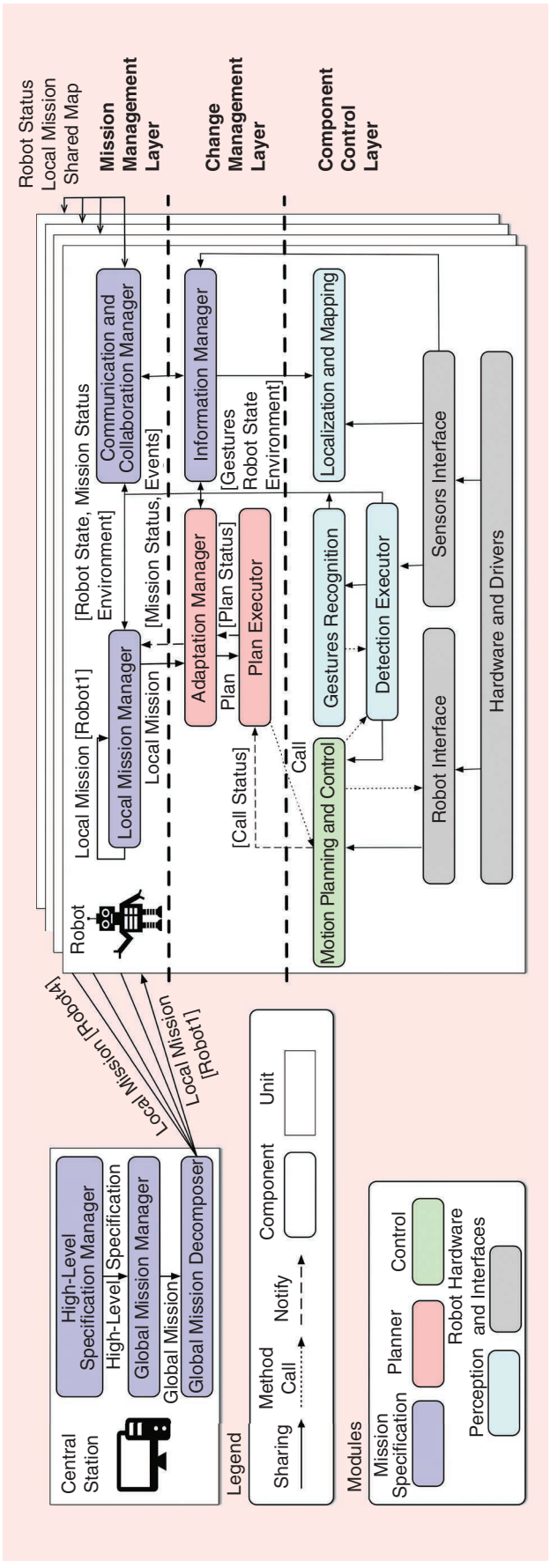


Figure 2. The components and interfaces of the SERA architecture.

properly adjusting its position and orientation with respect to the human or mobile manipulator.

Our focus is on the integration of perception, control, and planning modules that realize the successful cooperation of the heterogeneous agents in object manipulation and transportation tasks. The integration of these modules is facilitated by the adopted layered and component-based software architecture, called *SERA* [10] (a Self-adaptive dEcentralized Robotic Architecture). This architecture instantiates robotic applications by encapsulating different robotic functionalities within components. The components communicate in SERA by means of well-defined interfaces. SERA was developed to structure robotic applications formed by teams of (possibly) heterogeneous robots in a decentralized way during execution time. The robots must intercommunicate with the rest of the team and share data to achieve the global mission in a collaborative way. This intercommunication can be handled by the Robotic Operating System (ROS) infrastructure. An overview diagram of SERA that depicts components and their interfaces is presented in Figure 2.

Perception

The role of the perception module is to provide the robotic agents with the essential perceptual capabilities to accomplish the required missions. Three different methodologies have been developed for 1) multiple objects detection and tracking, 2) human detection and tracking, and 3) human posture estimation, as shown in Figure 3. The object detection and tracking algorithm is able to handle multiple objects and perform efficiently under occlusions using either red, green, blue (RGB) or RGB-depth (RGB-D) input. By employing a 3D model for each object, the algorithm initially learns the object appearance by detecting local features and registering them onto the surface of the 3D object model. Then, the features detection and matching procedure is performed, and, employing the random sample consensus method [11], the object pose is estimated considering the 3D model.

The novel hybrid human 3D body pose estimation method deployed [12] uses RGB-D input and relies on a deep neural network to get an initial 2D body pose. Using depth information from the sensor, a set of 2D landmarks on the body is transformed in 3D. Then, a multiple hypothesis tracker uses the obtained 2D and 3D body landmarks to estimate the 3D body pose using a gradient descent optimization scheme. Each human pose hypothesis is constructed using a different subset of the detected body landmarks. This way we safeguard against observation errors (i.e., misdetection of some of the landmarks) since we expect that some of these hypotheses will be free of misdetected landmarks.

Posture recognition builds upon the detected pose of each human. A given posture is detected by measuring the Euclidean distance between the template posture pose and the pose of each frame. A simple temporal filtering step is also used to ensure that the posture is detected in several consecutive frames before it is accepted as valid, thus avoiding spurious detection.

Control

A set of control modules is developed providing efficient solutions for robot navigation and manipulation tasks.

Navigation

Dealing with unstructured environments with unexpected obstacles (i.e., humans, other moving robots, and so forth) is essential for robot navigation. However, this type of environment is challenging for a robot to navigate because it must be capable of identifying and adapting to these changes. For this reason, the implemented control scheme guarantees collision avoidance for either dynamic or static obstacles and is responsible for regulating the motion of the platforms as they travel toward the different regions in the workspace. More specifically, the navigation methodology consists of two main algorithms integrated with the ROS 2D navigation stack [13].

The first one, which is used as a global planner, is based on harmonic potential fields [14]. This technique is selected due to its reduced computational requirement and the ability to handle large and complex workspaces. Additionally, it guarantees collision free navigation, with the goal configuration being the sole stable equilibrium. The proposed control scheme is based on the construction of a suitable transformation T_m, which maps 1) the robot's workspace $\mathcal{W}_{\text{mp}}$ to the punctured Euclidean plane, 2) the outer boundary $\mathcal{C}_{i,0}$ of the workspace to infinity, and 3) all obstacle boundaries $\mathcal{C}_{i,1}, \mathcal{C}_{i,2}, \ldots, \mathcal{C}_{i,N_{\text{obs}}^{\text{mp}}}$ and goal positions p_{mp}^{d} to distinct points $q_{\text{mp},o}^{i}, \forall i \in (1,2,\ldots,N_{\text{obs}}^{\text{mp}})$ and G, as shown in Figure 4. Thus, a feasible path can be computed that connects the current configuration of the robot with the desired one.

A time elastic band [15] approach is adopted for the local planner. The initial path generated by the harmonic maps technique is optimized with respect to minimization of trajectory execution time, obstacle avoidance, and compliance with kinodynamic constraints such as satisfying input and state constraints. Moreover, it complies with nonholonomic kinematic constraints by solving a sparse scalarized, multiobjective optimization problem.

Manipulation

In the proposed decentralized framework, the cooperative manipulation procedure among heterogeneous agents is crucial. In this context, the following control modalities have been implemented within the overall system architecture: 1) an object grasping algorithm that computes online the optimal grasping area, 2) a decentralized cooperative control scheme for automated loading tasks, and 3) a decentralized leader–follower cooperative object manipulation methodology.

The grasping method described in [16] has been selected due to its fast and robust performance. Moreover, it does not depend on offline training data or a 3D model of the object. The grasp planner relies only on the visible point cloud of the object and the characteristics of the robotic gripper such as

Figure 3. Three different methodologies that have been developed for perception modules. (a) Human detection and tracking. (b) Human postures. (c) Multiple object detection.

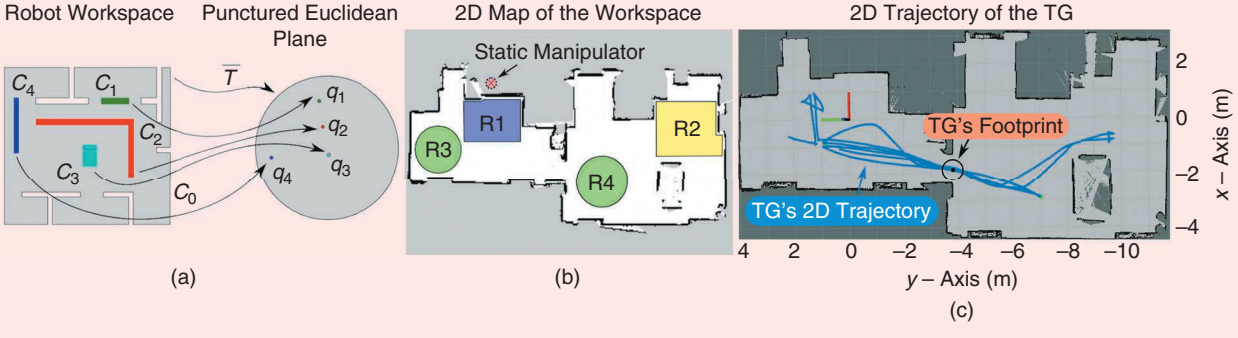


Figure 4. The navigation algorithm. (a) The transformation of a robot 2D map to a punctured disk, where T is the workspace transformation, C_i is the obstacles' boundary, and q_i is the corresponding distinct points on the punctured disk. (b) The R_i regions of interest in the 2D robots' map. (c) The 2D trajectory of the mobile manipulator on the aforementioned map. TG: mobile manipulator.

maximum opening, length, and height to calculate the optimal grasping area, as shown in Figure 5. This methodology is decoupled from the machine learning-based object detection algorithm due to the real-time feedback requirements during the reach-to-grasp phase. Moreover, when the robot is approaching the detected object, the object's point cloud noise acquired by the sensor is reduced, and its density is increased, making the aforementioned analytical methodology suitable for robust and reliable real-time grasping regions computation.

The initial path generated by the harmonic maps technique is optimized with respect to minimization of trajectory execution time, obstacle avoidance, and compliance with kinodynamic constraints such as satisfying input and state constraints.

The decentralized motion planning and control solution for the automated load exchange task among heterogeneous robots is described in a recent study [17]. More precisely, a motion planning algorithm based on probabilistic road maps calculates a connected graph $\mathcal{G}$. This graph consists of feasible configurations for the robotic system—which is holding the object to be loaded on a mobile platform—to facilitate the loading procedure, given the workspace constraints, its structure limitations, and the geometric characteristics of the mobile platform. The robotic system follows a feasible trajectory generated by a pathfinding algorithm employed on the computed graph $\mathcal{G}$ [Figure 6(a)]. Meanwhile, the mobile platform moves autonomously toward the object using the motion control scheme as described in the "Navigation" section. When the mobile platform reaches the loading area, the object is successfully placed on it. The previous presented method [17] is extended for three agents (a static manipulator, a mobile manipulator, and a mobile platform). Initially, the mobile manipulator and static manipulator are cooperatively grasping the object. Then, the mobile manipulator, acting as the leader, calculates a graph $\mathcal{G}$, consisting of connected feasible loading configurations of the object, taking into account the system limitations (i.e., static manipulator workspace limitations, obstacles, mobile platform's geometrical characteristics). Then, it calculates the optimal one and, performing a pathfinding algorithm, computes a feasible path that connects the initial object configuration with the desired one. Finally, the mobile manipulator leads the way to cooperatively transport the object with the static manipulator in a decentralized fashion by following the calculated path [Figure 6(b)]. The decentralized cooperative manipulation methodology is described in the following paragraphs.

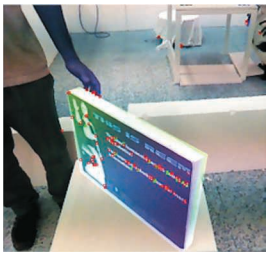
A leader–follower decentralized scheme is implemented for the cooperative object manipulation tasks. The follower is charged with the estimation of the desired trajectory utilizing a prescribed performance estimator following a similar

Grasping Algorithm Pseudocode

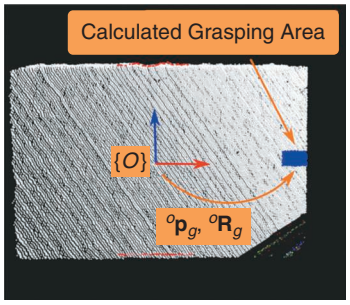
```
Require: The visible object's point cloud and gripper geometrical
         characteristics (maxOpening, height).

 1: procedure GraspAlgorithm (maxOpening, height)
 2:   while algorithm_enabled do
 3:     ObjectPoints ← GetObjectVisiblePoints()
 4:     ObjectPose ← GetObjectPose()
 5:     if size(ObjectPoints) > 0 then
 6:       ObjectPoints ← TransformPoints(ObjectPoints, ObjectPose)
 7:       contour ← ConcaveHullContour(ObjectPoints)
 8:       graspZone ← GraspableZone(contour, maxOpening)
 9:       ^O R_g, ^O p_g ← OptimalGrasp(graspZone, height)
10:       ^C R_g, ^C p_g ← TransformPoint (^g p_O, ObjectPose)
11:       return ^C R_g, ^C p_g
```

(a)



(b)



(c)

Figure 5. The optimal (a) grasping algorithm, (b) object detection, and (c) grasping area.

strategy as in [18]. In this context, we adopt a robust prescribed performance estimator that guarantees ultimate boundedness of the position and orientation estimation errors $e(t) \triangleq [e_p^\top(t), e_r^\top(t)]^\top = [e_1(t), \ldots, e_6(t)]^\top$ between the actual and the desired object configuration. The mathematical representation of the prescribed performance for each element is given by the following inequalities:

$$-\rho_j(t) < e_j(t) < \rho_j(t), \ \forall t \geq 0, j \in \{1, \ldots, 6\}, \qquad (1)$$

where $\rho_j(t)$ denotes the corresponding performance function that encapsulates the desired transient and steady-state performance specifications (e.g., convergence rate, maximum steady-state error). We choose as the exponential performance function the following one:

$$\rho_j(t) = (\rho_{j,0} - \rho_{j,\infty}) e^{-s_j t} + \rho_{j,\infty}, \qquad (2)$$

where the constant s_j dictates the exponential convergence rate, $\rho_{j,\infty}$ denotes the ultimate bound at the steady state, and $\rho_{j,0}$ is chosen to satisfy $p_{j0} > |e_j(0)|$. Hence, following the prescribed performance control methodology provides the desired motion intention trajectory profile. Notice that the estimation law is capable of estimating position, velocity, and acceleration based on only the actual position and velocity measurements. Then, the estimated trajectory is fed in an impedance/admittance control scheme to facilitate the transportation/manipulation task and limit the interaction wrenches. As the method relies exclusively on the robot's force/torque, position, velocity measurements and no explicit data are exchanged online between the robots, the object dynamics—which are considered known—along with the estimated acceleration are employed to compute the leader's applied wrench. A leader could be considered either a human agent (Figure 7) or a robotic agent [Figure 6(b)]. An abstraction of the proposed methodology is depicted in Figures 8 and 9 for the static manipulator and mobile manipulator, respectively.

High-Level Planning

Multiagent task planning of heterogeneous robots is also of utmost importance when it comes to coordinating robotic teams. Thus, a systematic, real-time, decentralized methodology to accomplish complex mission specifications given to a team of heterogeneous robots is employed.

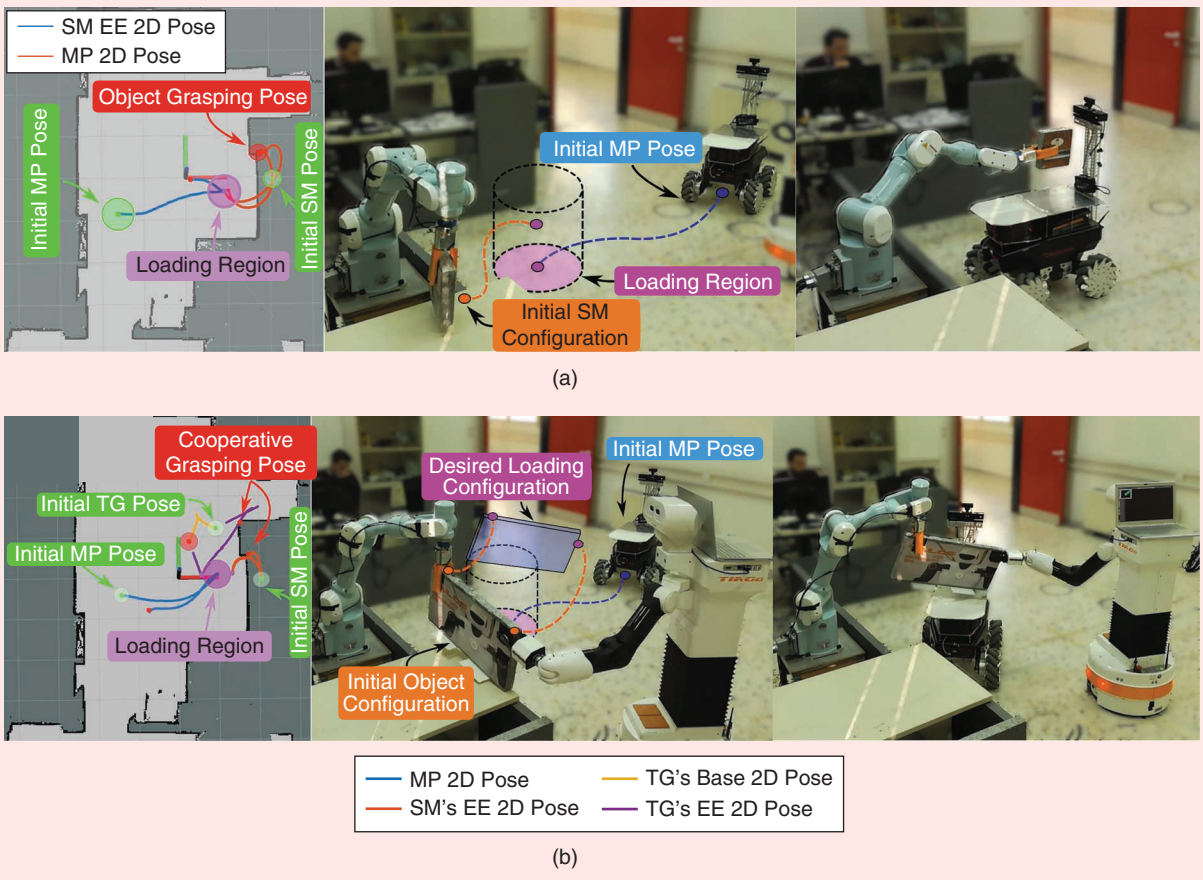


Figure 6. The cooperative loading procedure. (a) The mobile platform and static manipulator. (b) The mobile platform, mobile manipulator and static manipulator. MP: mobile platform; SM: static manipulator.

Figure 7. The human–robot cooperative object transportation. (a) Cooperative object grasping. (b) Cooperative object transportation. (c) Gesture recognition. F/T: force/torque.

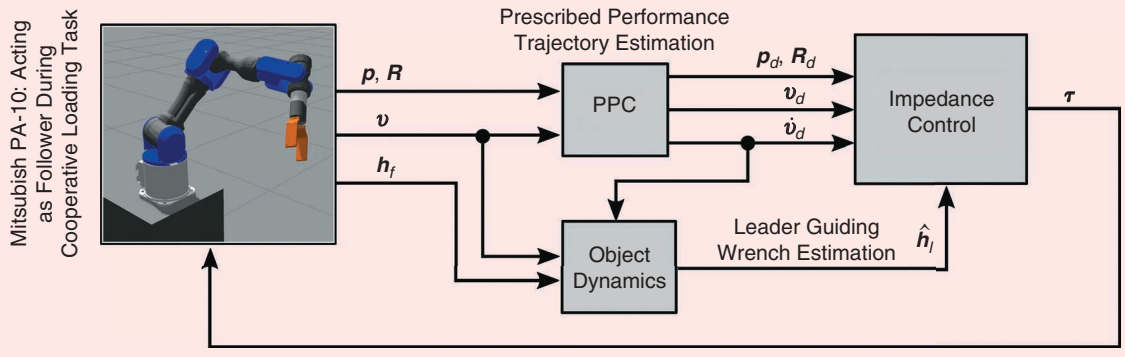


Figure 8. A block diagram of the follower robotic agent's control scheme for the desired trajectory estimation and tracking. PPC: prescribed performance control.

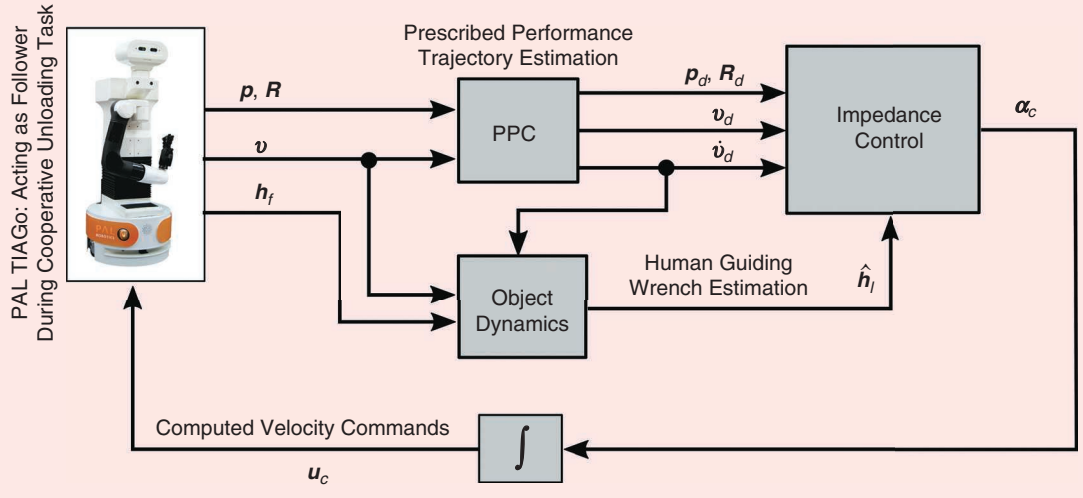


Figure 9. A block diagram of the mobile manipulator control scheme during the human–robot cooperative transportation procedure.

Graphical User Interface

To support users when specifying missions for their robotic applications, we strived to raise the levels of abstractions of our framework. To this purpose, we identified and formalized a catalog of mission-specification patterns containing recurrent specifications of missions (http://roboticpatterns.com/). Each pattern in the catalog is formulated both in structured English and in linear temporal logic (LTL). Then, to enable the specification of more complex and sophisticated missions, we created the domain-specific language Promise [19]. Promise (https://promisedsl.wixsite.com/promise) uses the mission specification patterns as basic building blocks and enables the specification of a mission via the use of composition operators. According to the SERA [10] architecture described previously, a global mission specification is subsequently decomposed into local missions, which are then individually forwarded to the robots. Thanks to Promise and the specification patterns, a mission can be specified in a user-friendly and graphical way and can then be automatically translated into LTL formulations.

Temporal Logic Formulation

The planning module follows the ideas described in [20] and [21] and consists of two main functionalities: first, it synthesizes the motion and action plan that fulfills an assigned task, which might be partially infeasible initially; then, it incorporates new features in the workspace model and revises the discrete plan accordingly.

The basic ingredients of an LTL formula are a set of atomic propositions and several Boolean and temporal operators, which are formed accordingly. More specifically, we model the motion and actions for the robots as finite transition systems $\mathcal{M}_i, \forall i \in (1,2,\ldots,N_R)$ as follows. The internal states of the robots, for example, "The robot has grasped object 1," are represented by sets of Boolean variables $\Psi_i, \forall i \in (1, 2, \ldots, N_R)$. Then, we model the action maps of the robots as the finite transition systems $\mathcal{B}_i, \forall i \in (1, 2, \ldots, N_R)$, which are based on precondition and effect functions for the actions (e.g., the robot can grasp an object in a region only if the object is in that region). Based on the aforementioned maps, we model the coupled behavior of each robot as the coupled transition system $\mathcal{R}_i = \mathcal{M}_i \times \mathcal{B}_i$, which is the product of the motion and action tuples. More details can be found in [21].

To find a plan over $\mathcal{R}$ that satisfies the assigned task, we employ standard techniques from formal verification methodologies. First, the assigned task is expressed as an LTL formula φ, which is then converted to a nondeterministic Büchi automaton A_φ. Then, we construct the product $A_P = \mathcal{R} \times A_\varphi$, and, using graph search algorithms, we obtain a least-violating plan over $\mathcal{R}$. The plan is least-violating in the sense that it satisfies most of the formula φ, given the initially partially known workspace. While the plan is executed, the robot obtains new information based on its sensing, and updates its knowledge about the workspace and hence the transition system $\mathcal{R}$. A new plan is then computed to improve the satisfiability of the assigned task, given the new workspace information.

Application

Experimental Setup and Scenario

We propose a single scenario that encapsulates all key concepts. More specifically, the scenario revolves around three robotic entities: 1) a mobile manipulator (PAL TIAGo), 2) a static manipulator (Mitsubishi PA-10), and 3) a mobile platform (Summit XL-HL), interacting with each other as well as with objects of various sizes that can be grasped, the environment, and the humans. In this scenario, objects of different sizes are loaded autonomously on top of the mobile platform. The latter transports the loaded object into a different room, where the unloading procedure is realized. Task allocation, planning, and the role of the engaged robotic entities are modified accordingly for heavy and light objects.

Thanks to Promise and the specification patterns, a mission can be specified in a user-friendly and graphical way and can then be automatically translated into LTL formulations.

During the scenario, the static manipulator is located next to a table with objects on top. The manipulator grasps the objects, one at a time. If the object is heavy, the mobile manipulator is additionally called for help, and the grasping and loading procedure is followed cooperatively by the two robots. At the same time, the mobile platform travels autonomously toward the loading area, where the object is loaded on top of the mobile platform. Next, the mobile platform travels to another room where a human waits in the unloading area. If the object is heavy, the human also calls the mobile manipulator for help via an appropriate posture, which is performed in front of the mobile platform. Together, the human and the mobile manipulator grasp the object and cooperatively unload it on top of a table. When the procedure is completed, the mobile manipulator returns to its surveillance tasks until another loading procedure is initiated. When the object is light, the same procedure is followed but without engaging the mobile manipulator, which is left operating in surveillance mode. A video of the demonstration is available at https://youtu.be/q7dMLawf0y0.

Experimental Results

One of our industrial partners, Bosch, provided benchmarks for measuring the research success driven by market and product needs. Within the project, certain measurable quantities are evaluated to track the success of

the research activities. Following the objectives of the project, the three main criteria for evaluation were 1) flexibility, 2) robustness, and 3) efficiency. Objectives relating to these three goals are defined separately for each of the modules described in the "System Components" section, and the measures of success for these key objectives are specified in Table 1.

Perception Module

The perception module performs the tasks of object pose estimation, human body pose estimation, and gestures recognition. The object pose estimation system was trained to detect the two objects shown in Figure 3. Training for each object requires a 3D model and some annotated RGB-D frames from different viewpoints. For the purpose of this scenario, the system was able to successfully detect and track in 3D the two objects that were involved. The method can handle objects of various sizes and shapes, given that they have some appropriate textured faces.

Quantitative results of the human and object pose estimation modules' accuracy require ground truth pose (e.g., provided by a motion capture system) that was not available during the experiments. However, we provide test results of the proposed methods, with data sets that provide ground truth pose measurements. Table 2 presents the accuracy and the computation cost of the perception algorithms.

Control Module

The grasping procedure runs on both heterogeneous manipulators (TIAGo and PA-10) for two different object types (heavy and light) to realize the loading and unloading tasks. The ratios of successfully grasped objects that were recorded during the experiments are depicted in Table 3. Moreover, the errors of the six DoFs between the robots' end effectors and the calculated grasping poses as recorded during three different successful grasps are shown in Figure 10.

Figure 6(a) depicts the initial pose of the static manipulator and the mobile platform with green colored circles, the mobile platform's obstacle-free path with a blue line, and the calculated loading region with a magenta-colored circle. Similarly, Figure 6(b) depicts the initial pose of the static manipulator, the mobile manipulator, and the mobile platform with green-colored circles, the mobile platform's obstacle-free path with a blue line, the calculated loading region with a magenta-colored circle, and the final mobile manipulator's base pose with the red line circle. Required time and the ratio of successful loading tasks using the light and heavy objects are shown in Table 4.

In Figures 11 and 12, the actual interaction forces are depicted to support the fact that the trajectory estimation along with the impedance control succeeded in the cooperative execution of a common trajectory while keeping the interaction wrenches limited. Moreover, the force estimation depicted in Figure 11 reveals that, as long as the object dynamics are approximately known, the wrench of the leader can be effectively estimated. Thus, explicit communication between the robots is not necessary for either the leader's trajectory or the wrench. As Figure 12 shows, the follower's (TIAGo) end-effector errors between the current pose and the estimated remain generally close to zero during the task. Thus, the values of the wrench applied by the leader on the object are, in general, small enough, making the human effort inconsiderable during the transportation mission.

Planning Module

The time taken for a new task to be set up is less than 0.1 s; that for the plan derivation satisfying the task is less than 0.1 s, and that for plan synthesis and reconfiguration is also less than 0.1 s.

Table 1. Measures of success for the proposed architecture.

	Flexibility	Robustness	Efficiency
Control module	Variability of object types and transfer goals	Ratio of successful object grasping and loading/unloading missions	Time for cooperative loading and unloading procedures
Perception module	Number/types of objects that can be recognized	Accuracy of human and object detection/pose estimation under occlusions	Computational cost
Planning module	Number of different workspace configurations	Number and type of external events that can be handled	Time required for plan synthesis

Table 2. Perception module: human and object detection/pose estimation evaluation.

	Pose Accuracy	Computational Cost
Object detection	~1 cm	~ 8 fps (on a 2.8 GHz Intel Core i7-7700HQ CPU)
Human detection	~6.6 cm	~ 5 fps (on a Nvidia GeForce GTX 1070)

Table 3. Ratio of successful object grasping.

	Heavy Box	Light Box
PA-10	91% (11 successful out of 12 attempts)	83% (10 successful out of 12 attempts)
TIAGo	80% (8 successful out of 10 attempts)	Not needed

The time taken for the execution of the task depends on the respective continuous control algorithms. Typically, the time taken for navigation among the regions of interest as well as single or cooperative loading and unloading is ~10–30 s. The planner can handle any number of workspace and initial configurations.

The execution of the plan might be jeopardized by significant disturbances in the state feedback. Modeling errors might also affect the plan, for instance, in cases in which the robots cannot execute an action to which they are modeled. However, no such events occurred in the tested scenario, the plan was successfully executed despite of the noisy measurements, and plan reconfiguration was successfully performed in the modeled cases of new environment information.

Discussion

The proposed decentralized framework was tested in a scenario that included a sufficient number of tasks to assess the

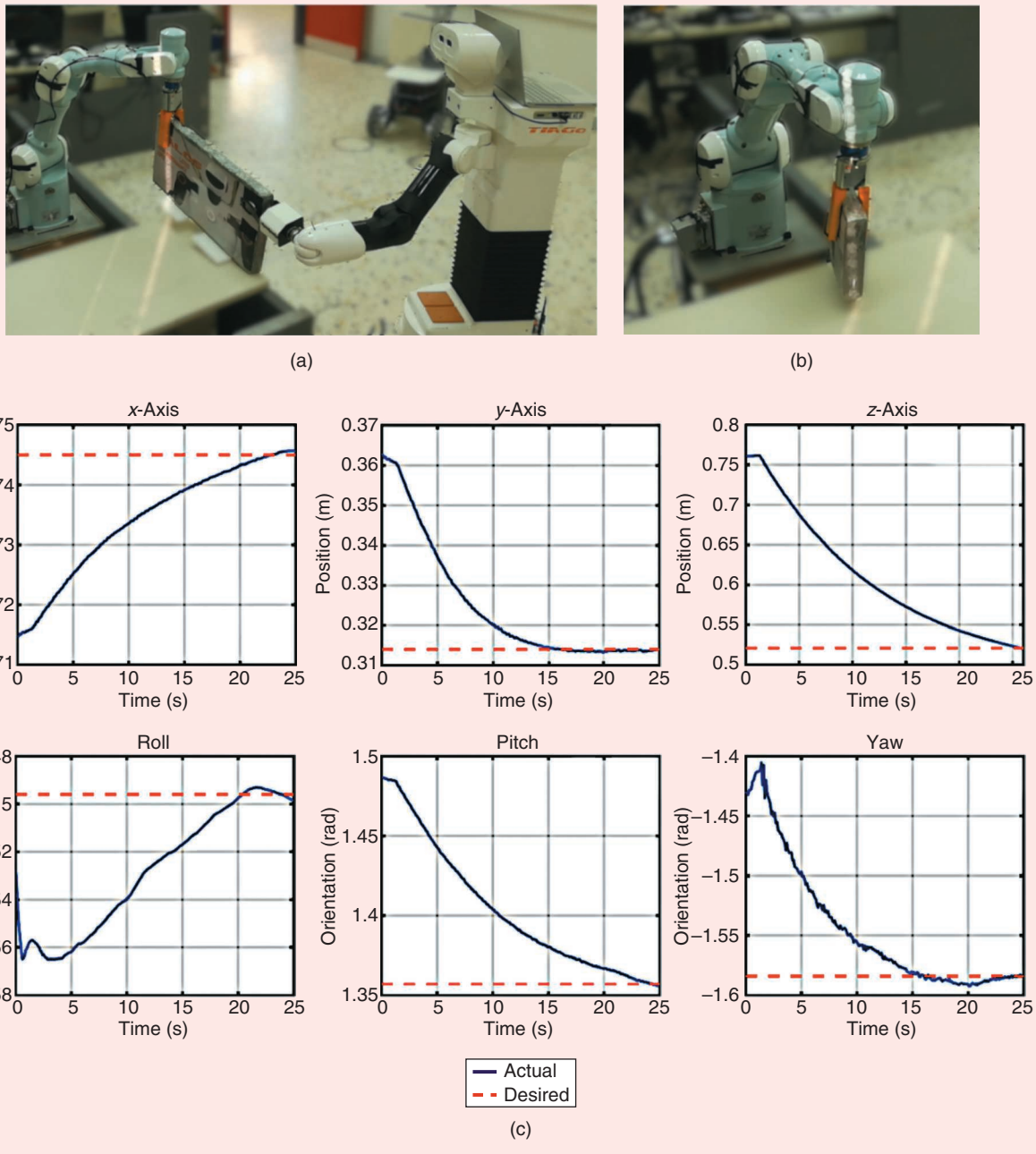


Figure 10. The recorded errors of the six DoFs between the robots' end effectors and the calculated grasping poses. (a) TIAGo and PA-10 performing heavy object (HB) grasping task. (b) PA-10 performing light object (LB) grasping task. (c) TIAGo end-effector pose errors performing LB grasping task.

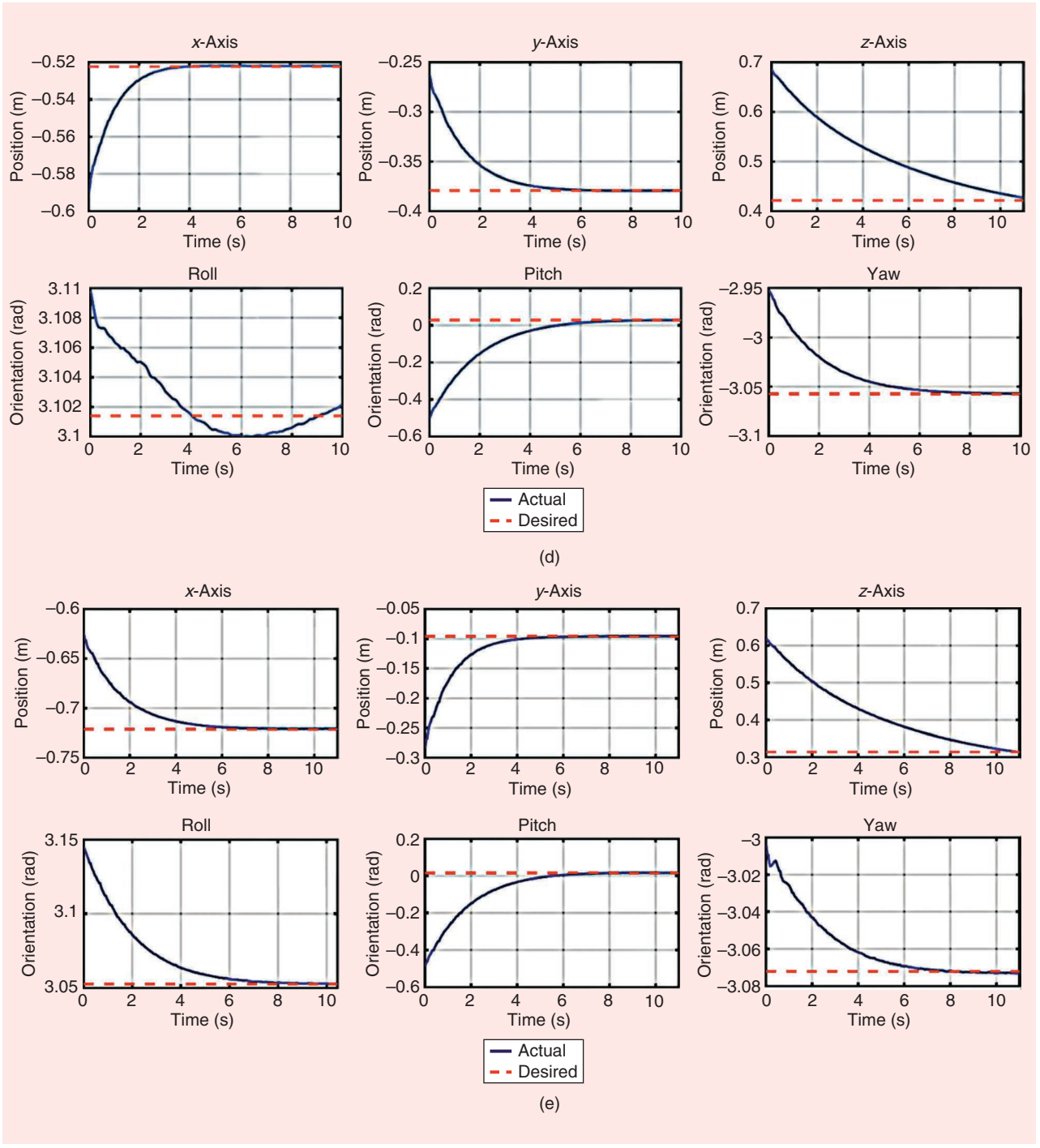


Figure 10. *(Continued)* (d) PA-10 end-effector pose errors performing HB grasping task. (e) PA-10 end-effector pose errors performing LB grasping task.

Table 4. Ratio and time needed for successful cooperative loading procedures.

	Ratio of Success	Time Needed
PA-10 and Summit	90% (9 successful out of 10 attempts)	38.5 s
PA-10, TIAGo, and Summit	83% (10 successful out of 12 attempts)	48.2 s

performance of each individual component as well as the performance of the overall architecture. However, the multiagent system consisted of only one agent per category (i.e., mobile platform, static, mobile manipulator, and a human). Therefore, even if the proposed framework is by design scalable, the actual scalability has not yet been efficiently demonstrated. In the future, we intend to address scalability by adding more agents, both robotic and humans; objects of various geometries; and an expanded

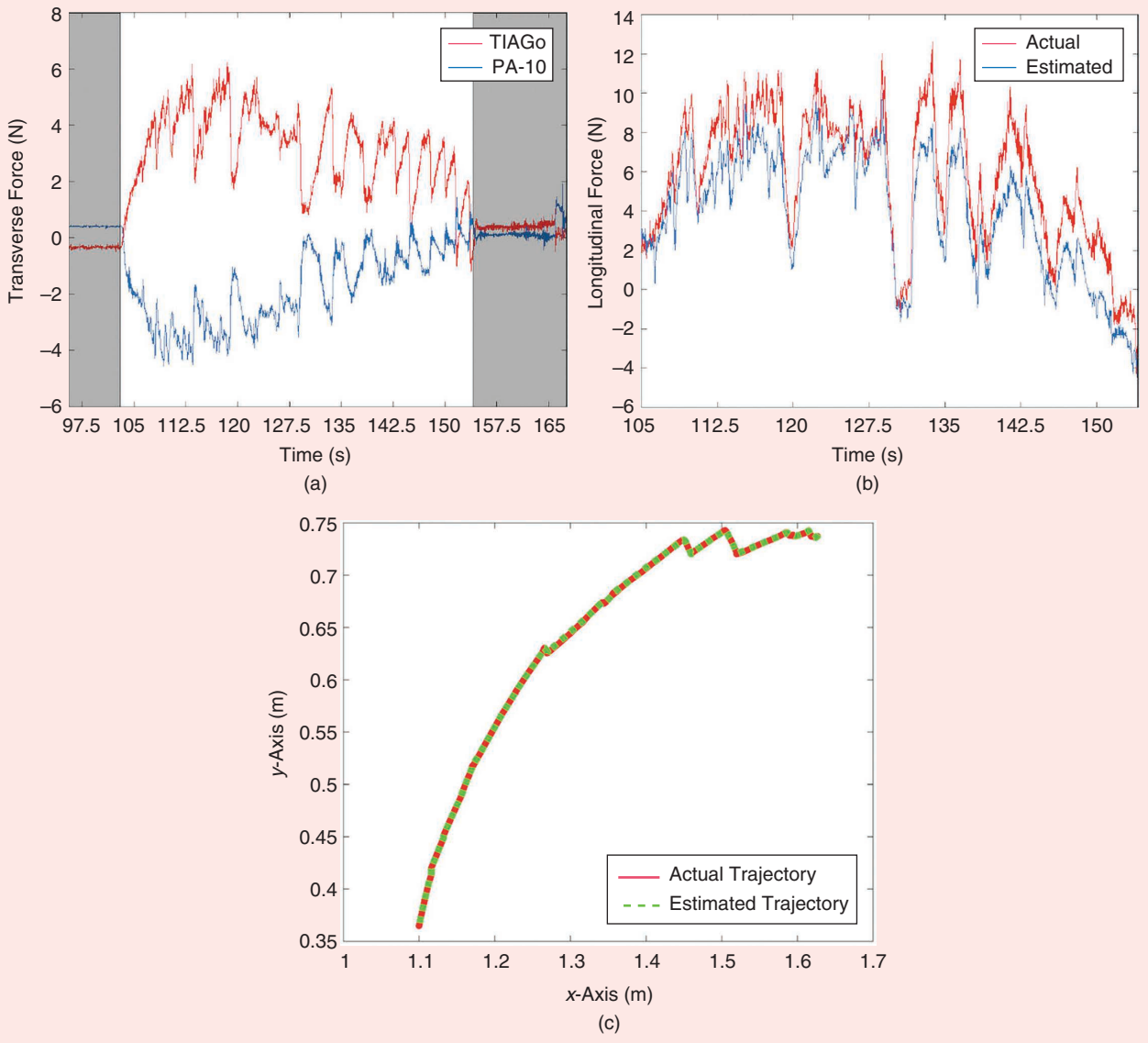


Figure 11. The force/torque sensing-estimation and trajectory estimation during the cooperative loading task. (a) The interaction forces during the cooperative transportation task. The leader's (mobile manipulator) interaction force as estimated by the follower (static manipulator). (b) The actual ones are depicted. (c) The leader-estimated 2D trajectory and the actual one.

set of high-level tasks. Moreover, an important issue that we have not yet addressed is the online reconfiguration of planning in the presence of actuation or sensor faults, which also mandates an increase in the number of involved agents. Additionally, a significant attribute that we intend to incorporate in the future pertains the aspects of cognitive safety in human–robot collaboration tasks. More specifically, we aim to tackle issues such as human situation awareness and motion intention (inside the workspace) as well as inherited compliant behavior during cooperative manipulation tasks. In this way, we will be able to ensure that robots and humans can safely and effectively coexist in a real dynamic environment.

Conclusions

In this work, we propose a decentralized framework for the efficient cooperation of heterogeneous robotic agents and humans in manipulation and transportation tasks within semistructured workspaces. This framework consists of separate modules responsible for perception, motion, and manipulation control, as well as high-level planning. We evaluated the integrated system in a multitasking scenario involving various heterogeneous robots and humans for the cooperative loading/unloading and transportation of objects. A series of metrics such as accuracy, ratio of success, and computational cost justify the robustness, flexibility, and efficiency of the overall framework.

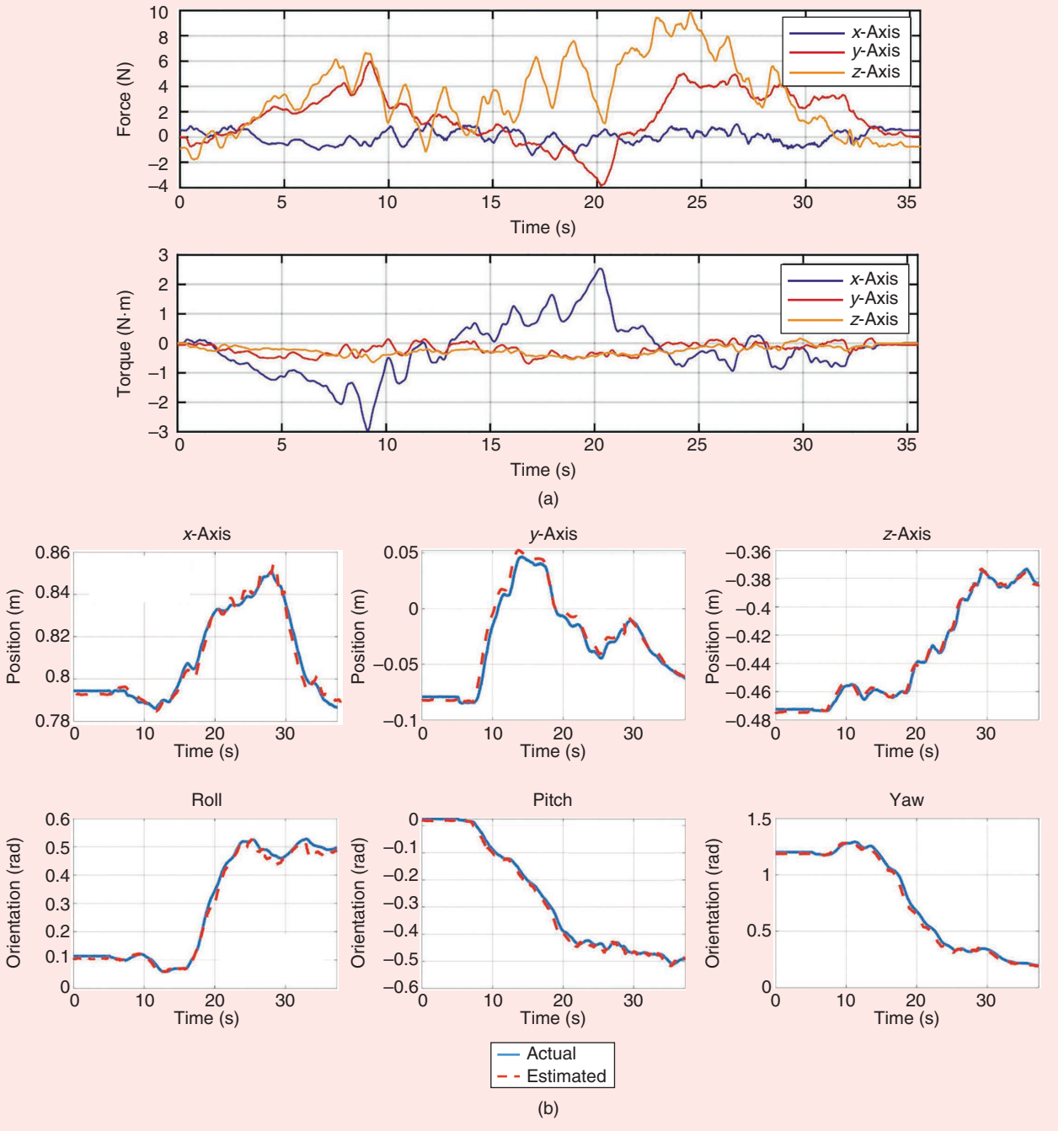


Figure 12. The force/torque sensing and trajectory estimation in a cooperative human–TIAGo unloading task between the current pose and the estimate. (a) The TIAGo force and torque measurement. (b) The estimated desired and actual pose of the TIAGo's end-effector.

Acknowledgment

This work was supported by the European Community through the project Co4Robots (H2020-731869). The authors would like to thank the rest of the Co4Robots Consortium for all of their efforts during the project.

References

[1] Z. G. Saribatur, E. Erdem, and V. Patoglu, "Cognitive factories with multiple teams of heterogeneous robots: Hybrid reasoning for optimal feasible global plans," in *Proc. 2014 IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS 2014)*, pp. 2923–2930. doi: 10.1109/IROS.2014.6942965.

[2] S. M. LaValle, *Planning Algorithms*. Cambridge, U.K.: Cambridge Univ. Press, 2006.

[3] J. Alonso-Mora, J. A. DeCastro, V. Raman, D. Rus, and H. Kress-Gazit, "Reactive mission and motion planning with deadlock resolution avoiding dynamic obstacles," *Auton. Robots*, vol. 42, no. 4, pp. 801–824, 2018. doi: 10.1007/s10514-017-9665-6.

[4] C. K. Verginis, Z. Xu, and D. V. Dimarogonas, "Decentralized motion planning with collision avoidance for a team of uavs under

high level goals," in *Proc. 2017 IEEE Int. Conf. Robot. Automat. (ICRA)*, pp. 781–787. doi: 10.1109/ICRA.2017.7989096.
[5] Z. Liu, Y. Yamauchi, S. Kijima, and M. Yamashita, "Team assembling problem for asynchronous heterogeneous mobile robots," *Theoretical Comput. Sci.*, vol. 721, pp. 27–41, Apr. 2018. doi: 10.1016/j.tcs.2018.01.009.
[6] H. Tanner, S. Loizou, and K. Kyriakopoulos, "Nonholonomic navigation and control of cooperating mobile manipulators," *IEEE Trans. Robot.*, vol. 19, no. 1, pp. 53–64, 2003. doi: 10.1109/TRA.2002.807549.
[7] W. C. Dickson, R. H. Cannon, and S. M. Rock, "Decentralized object impedance controller for object/root-team systems: Theory and experiments," in *Proc. IEEE Int. Conf. Robot. Automat.*, 1997, pp. 3589–3596.
[8] K. Kosuge, T. Oosumi, and H. Seki, "Decentralized control of multiple manipulators handling an object in coordination based on impedance control of each arm," in *Proc. IEEE Int. Conf. Intell. Robots Syst.*, 1997, pp. 17–22. doi: 10.1109/IROS.1997.648976.
[9] A. Marino and F. Pierri, "A two stage approach for distributed cooperative manipulation of an unknown object without explicit communication and unknown number of robots," *Robot. Auton. Syst.*, vol. 103, pp. 122–133, May 2018. doi: 10.1016/j.robot.2018.02.007.
[10] S. García, C. Menghi, P. Pelliccione, T. Berger, and R. Wohlrab, "An architecture for decentralized, collaborative, and autonomous robots," in *Proc. IEEE Int. Conf. Softw. Architecture (ICSA)*, 2018, pp. 7500–7509. doi: 10.1109/ICSA.2018.00017.
[11] M. Lourakis and X. Zabulis, *Model-Based Pose Estimation for Rigid Objects*. Berlin: Springer-Verlag, 2013, pp. 83–92.
[12] A. Makris and A. A. Argyros, "Robust 3d human pose estimation guided by filtered subsets of body keypoints," in *Proc. 16th Int. Conf. Mach. Vision Appl.*, 2019, pp. 1–6. doi: 10.23919/MVA.2019.8757907.
[13] "ROS navigation stack." Robot Operating System. http://wiki.ros.org/navigation (accessed Jan. 2020).
[14] P. Vlantis, C. Vrohidis, C. P. Bechlioulis, and K. J. Kyriakopoulos, "Robot navigation in complex workspaces using harmonic maps," in *Proc. IEEE/RSJ Int. Conf. Robot. Automat. (ICRA)*, 2018, pp. 1726–1731. doi: 10.1109/ICRA.2018.8460695.
[15] C. Rösmann, F. Hoffmann, and T. Bertram, "Integrated online trajectory planning and optimization in distinctive topologies," *Robot. Auton. Syst.*, vol. 88, pp. 142–153, Feb. 2017. doi: 10.1016/j.robot.2016.11.007.
[16] M. Logothetis, G. C. Karras, S. Heshmati-Alamdari, P. Vlantis, and K. J. Kyriakopoulos, "A model predictive control approach for vision-based object grasping via mobile manipulator," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct 2018, pp. 1–6. doi: 10.1109/IROS.2018.8593759.
[17] M. Logothetis, P. Vlantis, C. Vrohidis, G. C. Karras, and K. J. Kyriakopoulos, "A motion planning scheme for cooperative loading using heterogeneous robotic agents," in *Proc. 2019 IEEE/RSJ Int. Conf. Robot. Automat. (ICRA)*, pp. 9660–9666. doi: 10.1109/ICRA.2019.8794323.
[18] C. Mavridis, K. Alevizos, C. P. Bechlioulis, and K. J. Kyriakopoulos., "Human-robot collaboration based on robust motion intention estimation with prescribed performance," in *Proc. 2018 Eur. Control Conf.,* Limassol, Cyprus, vol. 1, pp. 249–254. doi: 10.23919/ECC.2018.8550610.
[19] S. García, P. Pelliccione, C. Menghi, T. Berger, and T. Bures, "High-level mission specification for multiple robots," in *Proc. 12th ACM SIGPLAN Int. Conf. Software Language Eng.*, 2019, pp. 127–140. doi: 10.1145/3357766.3359535.
[20] M. Guo and D. V. Dimarogonas, "Multi-agent plan reconfiguration under local LTL specifications," *Int. J. Robot. Res.*, vol. 34, no. 2, pp. 218–235, 2015. doi: 10.1177/0278364914546174.
[21] M. Guo, K. H. Johansson, and D. V. Dimarogonas, "Motion and action planning under LTL specifications using navigation functions and action description language," *IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 2013, pp. 240–245. doi: 10.1109/IROS.2013.6696359.

Michalis Logothetis, Control Systems Lab, Department of Mechanical Engineering, National Technical University of Athens, Zografou, 15780, Greece. Email: logothm@gmail.com.

George C. Karras, Department of Computer Science and Telecommunications, University of Thessaly, 35100, Lamia, 35100, Greece, and Control Systems Lab, Department of Mechanical Engineering, National Technical University of Athens, Zografou, 15780, Greece. Email: gkarras@uth.gr.

Konstantinos Alevizos, Control Systems Lab, Department of Mechanical Engineering, National Technical University of Athens, Zografou, 15780, Greece. Email: kosalevizos@mail.ntua.gr.

Christos K. Verginis, School of Electrical Engineering and Computer Science, KTH Royal Institute of Technology, Stockholm, SE-100 44, Sweden. Email: chrisverginis@gmail.com.

Pedro Roque, School of Electrical Engineering and Computer Science, KTH Royal Institute of Technology, Stockholm, SE-100 44, Sweden. Email: padr@kth.se.

Konstantinos Roditakis, Institute of Computer Science, FORTH, Heraklion, GR-700 13, Greece. Email: croditak@ics.forth.gr.

Alexandros Makris, Institute of Computer Science, FORTH, Heraklion, GR-700 13, Greece. Email: amakris@ics.forth.gr.

Sergio García, Chalmers University of Technology, Gothenburg, 41756, Sweden. Email: sergio.garcia@gu.se.

Philipp Schillinger, Bosch Center for Artificial Intelligence, Renningen, 71272, Germany. Email: Philipp.Schillinger@de.bosch.com.

Alessandro Di Fava, PAL Robotics, Barcelona, 08005, Spain. Email: alessandro.difava@pal-robotics.com.

Patrizio Pelliccione, Chalmers University of Technology, Gothenburg, L'Aquila AQ, 67100, Sweden. Email: patrizio.pelliccione@gu.se.

Antonis Argyros, Institute of Computer Science, FORTH, Heraklion, GR-700 13, Greece. Email: argyros@ics.forth.gr.

Kostas J. Kyriakopoulos, Control Systems Lab, Department of Mechanical Engineering, National Technical University of Athens, Zografou, 15780, Greece. Email: kkyria@mail.ntua.gr.

Dimos V. Dimarogonas, School of Electrical Engineering and Computer Science, KTH Royal Institute of Technology, Stockholm, SE-100 44, Sweden. Email: dimos@kth.se.