A Decentralized Framework for Efficient Cooperation of Heterogeneous Robotic Agents

Michalis Logothetis[1], George C. Karras[2,1], Konstantinos Alevizos[1], Christos K. Verginis[3], Pedro Roque[3], Konstantinos Roditakis[4], Alexandros Makris[4], Sergio García[5], Philipp Schillinger[6], Alessandro Di Fava[7], Patrizio Pelliccione[5], Antonis Argyros[4], Kostas J. Kyriakopoulos[1], Dimos V. Dimarogonas[3]

In social and industrial facilities of the future like hospitals, hotels, and warehouses, teams of robots will be deployed to assist humans accomplish everyday tasks such as 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 in order to achieve various complex tasks (e.g. cooperative parts assembly in automotive industry, or loading and unloading of palettes in warehouses) that require collaborative actions with each other and with human operators (Figure 1).

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 robots dynamic environment. Decentralization in multi-agent robotic systems is of utmost importance, since it provides flexibility, scalability and fault-tolerance capabilities. In this work, we present the architecture of the decentralized framework developed within the context of EU Project Co4Robots and its application in a multi-tasking collaboration scenario involving various heterogeneous robots and humans.

I. Background

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 multi-robot systems is a rather tricky under tacking owing to the different kinematic and sensing capabilities of each robot. One issue of utmost importance in coordination of robotic teams is the multi-agent task planning and control. Towards this direction, significant efforts have been devoted in the last decades resulting in a number of high complexity algorithms [2].

[1]Control Systems Lab, Department of Mechanical Engineering, National Technical University of Athens, 9 Heroon Polytechniou Street, Zografou 15780, Greece. Email: {logothm,kosalevizos,kkyria}@mail.ntua.gr

[2]Dept. of Computer Science and Telecommunications, University of Thessaly, 3rd Km Old National Road Lamia Athens, 35100, Lamia, Greece. Email: gkarras@uth.gr

[3]School of Electrical Engineering and Computer Science, KTH Royal Institute of Technology, SE-100 44, Stockholm, Sweden. Email: {cverginis,padr,dimos}@kth.se

[4]Institute of Computer Science, FORTH, GR-700 13, Heraklion, Greece. Email: {croditak,amakris,argyros}@ics.forth.gr

[5]Chalmers | University of Gothenburg, Gothenburg, Sweden. Email: { sergio.garcia,patrizio.pelliccione}@gu.se

[6]Bosch Center for Artificial Intelligence, Germany Email: Philipp.Schillinger@de.bosch.com

[7]PAL Robotics, Spain Email: alessandro.difava@pal-robotics.com

Fig. 1: Cooperative Tasks of Heterogeneous Agents

A standard classification that arises in multi-agent 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 at a great deal 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 are proven computationally expensive. On the other hand, decentralized planning, reduces significantly the computational complexity, [4]. A rather important issue in heterogeneous decentralized task-planning is the role assignment and the task allocation, where each agent's capabilities diverse and depend on its teammate and/or their mutual state [5].

From the control point of view, multi-robot cooperative object manipulation and transportation has 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 becomes large. On the other hand, decentralized control approaches usually depend on heavy inter-robot explicit communication and global offline knowledge of the desired task [7]. Nevertheless, in such tasks implicit inter-robot communication arises naturally as a side-effect of robot's physical interactions (e.g. the interaction forces between the object and the robot) which can be easily acquired by appropriate sensors attached on the robots [8]. However, limited studies have been conducted in cooperative object manipulation and transportation via hetero-

geneous robotic systems [9].

Hence, our work is motivated by the need of having multi-robot decentralized systems, where a significant amount of information can be implicitly acquired via physical interactions and processed locally, reducing in this way the need for exhaustive inter-robot explicit communication. More specifically, we present a complete decentralized framework consisting of: i) a set of perceptual algorithms that enables cooperating robots to estimate the state of their highly dynamic environment, ii) a set of control schemes appropriate for the mobility and manipulation capabilities of the considered robotic platforms, iii) a systematic real-time decentralized methodology to accomplish complex mission specifications given to a team of heterogeneous robots, and iv) the corresponding systematic integration of the above 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.

II. 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: i) a dexterous 7 DOF static manipulator able to perform loading and unloading actions of light and heavy objects, ii) a mobile manipulator to extend the motion flexibility and reachability of the overall framework, and iii) 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 properly adjusting its position and orientation with respect to the manipulator, 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 inter-communicate with the rest of the team and share data in order to achieve the global mission in a collaborative way. The intercommunication can be handled by the ROS' infrastructure. An overview diagram of SERA, which depicts components and their interfaces is presented in Figure 2.

A. Perception

The role of the perception module is to provide the robotic agents with the essential perceptual capabilities in order to accomplish the required missions. Three different methodologies have been developed for: i) *Multiple objects detection and tracking*, ii) *Human detection and tracking*, and iii) *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 RGB or 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 RANSAC method [11] the object pose is estimated considering the 3D model.

A novel hybrid human 3D body pose estimation method [12] that uses RGB-D input is deployed that 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 are 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 from 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 accepting it as valid, thus avoiding spurious detection.

B. Control

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

1) Navigation: Dealing with unstructured environments with unexpected obstacles (i.e. humans, other moving robots etc.) 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 it is responsible for regulating the motion of the platforms as they travel towards 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: i) the robot's workspace $\mathscr{W}_{mp}$ to the punctured Euclidean plane, ii) the outer boundary $\mathscr{C}_{i,0}$ of the workspace to infinity, iii) all obstacle boundaries $\mathscr{C}_{i,1}, \mathscr{C}_{i,2}, \ldots, \mathscr{C}_{i,N_{obs}^{mp}}$ and goal positions p_{mp}^{d} to distinct points $q_{mp,o}^{i}, \forall i \in (1,2,\ldots,N_{obs}^{mp})$

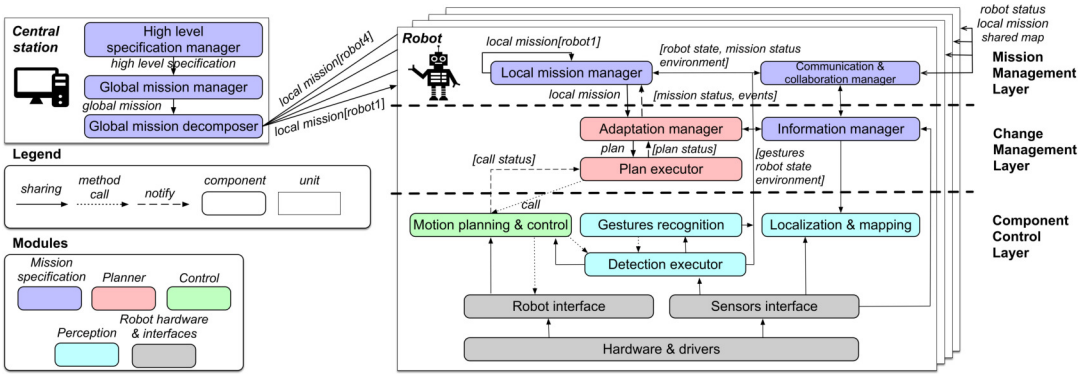


Fig. 2: The SERA architecture

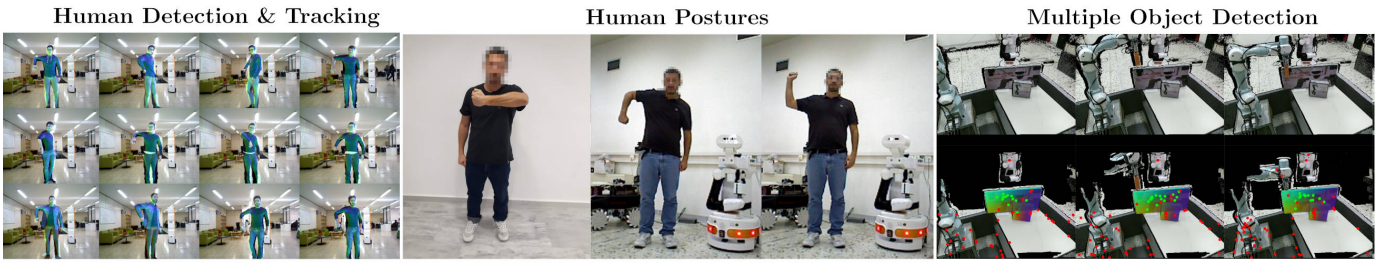


Fig. 3: Perception Modules

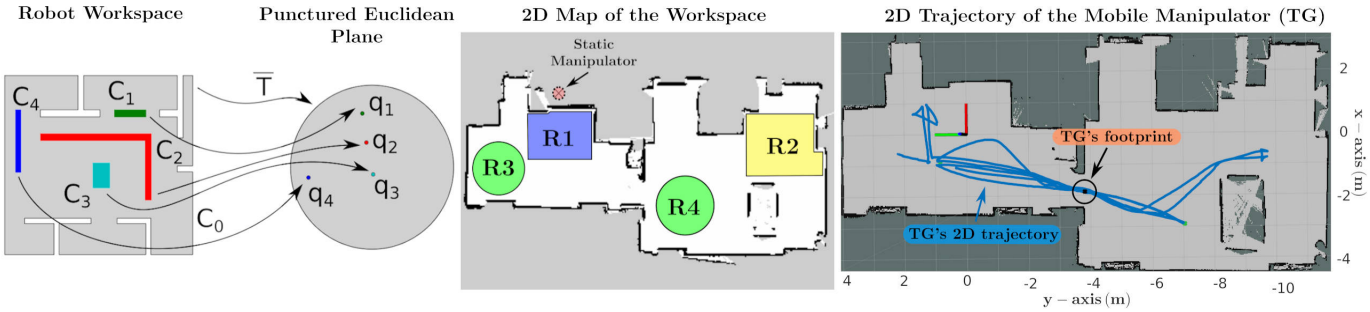


Fig. 4: Navigation Algorithm: The left figure depicts the transformation of a robot 2D map to a punctured disk, where T is the workspace transformation, C_i is the obstacles' boundary and q_i is the corresponding distinct points on the punctured disk. The middle figure shows the R_i regions of interest in the 2D robots' map. The 2D trajectory of the TG on the aforementioned map can be shown in the right figure.

Grasping Algorithm Pseudocode

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

```
procedure GraspAlgorithm(maxOpening, height)
    while algorithm_enabled do
        ObjectPoints ← GetObjectVisiblePoints()
        ObjectPose ← GetObjectPose()
        if size(ObjectPoints) > 0 then
            ObjectPoints ← TransformPoints(ObjectPoints, ObjectPose)
            contour ← ConcaveHullContour(ObjectPoints)
            graspZone ← GraspableZone(contour, maxOpening)
            ^O R_g, ^O p_g ← OptimalGrasp(graspZone, height)
            ^C R_g, ^C p_g ← TransformPoint(^B p_O, ObjectPose)
            return ^C R_g, ^C p_g
```

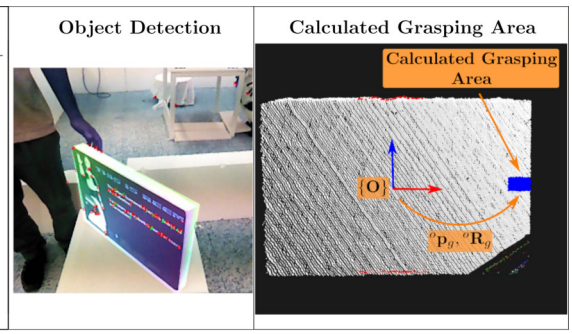
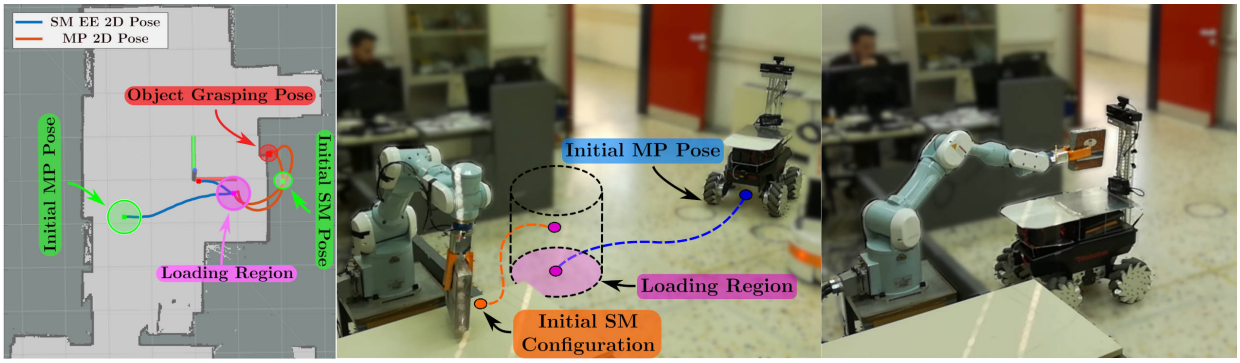


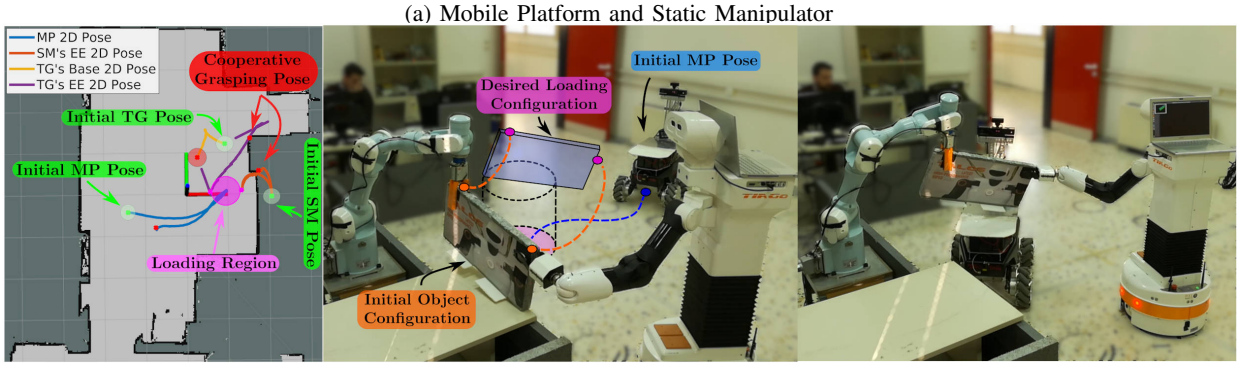
Fig. 5: Grasping Procedure

and q^d_{mp} as shown in Figure 4. Thus, a feasible path can be computed that connects the current configuration of the robot with the desired one.

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 minimizing the trajectory execution time, obstacle avoidance and compliance with kinodynamic constraints such as satisfying input and



(a) Mobile Platform and Static Manipulator



(b) Mobile Platform, Mobile Manipulator and Static Manipulator

Fig. 6: Cooperative Loading Procedure

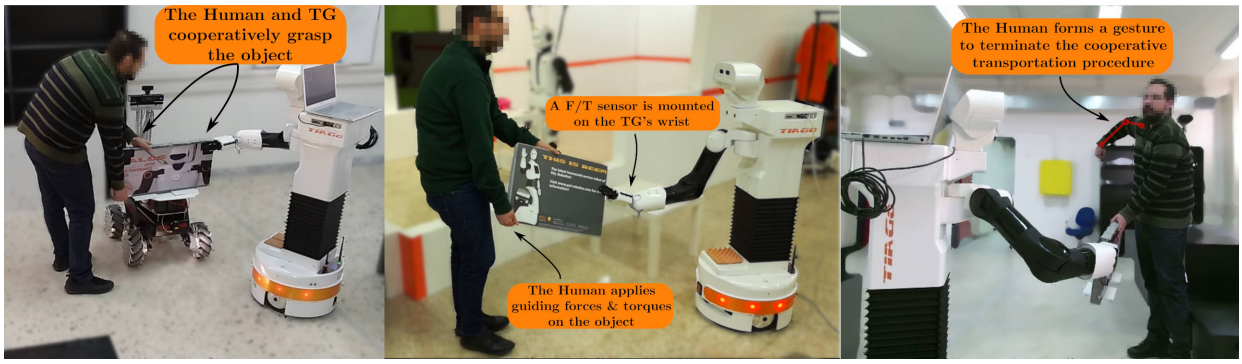


Fig. 7: Human-Robot Cooperative Object Transportation

state constraints. Moreover, it complies with non-holonomic kinematic constraints by solving a sparse scalarized multi-objective optimization problem.

2) 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: i) An *object grasping algorithm* which computes on-line the optimal grasping area, ii) a *decentralized cooperative control scheme for automated loading tasks*, iii) 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 off–line training data or 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 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 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 (PRMs) calculates a connected graph $\mathscr{G}$. This graph consists of feasible configurations for the robotic system—which is holding the object to be loaded on a mobile platform (MP)—to facilitate the loading procedure, given the workspace constraints, its structure limitations, and the geometric charac-

teristics of the mobile platform. The robotic system follows a feasible trajectory generated by a path finding algorithm which is employed on the computed graph $\mathscr{G}$ (Figure 6a). In the meanwhile, the mobile platform moves autonomously towards the object using the motion control scheme described in Sec. II-B1. When the mobile platform reaches the loading area, the object is successfully placed on it. The previous presented method in [17] is extended for three agents (a static manipulator, a mobile manipulator and a mobile platform). Initially, the mobile manipulator (TG) and the static manipulator (SM) are cooperatively grasping the object. Then, the TG, acting as the leader, calculates a graph $\mathscr{G}$ consists of connected feasible loading configurations of the object taking into account the system limitations (i.e., SM workspace limitations, obstacles, mobile platform's geometrical characteristics). Then, it calculates the optimal one and performing a path finding algorithm computes a feasible path that connects the initial object configuration with the desired one. Finally, TG leads the way to cooperatively transport the object with the SM in a decentralized fashion by following the calculated path (Figure 6b). The decentralized cooperative manipulation methodology is described below.

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 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 $e(t) \triangleq \left[e_p^\top(t), e_r^\top(t)\right]^\top = [e_1(t), \ldots, e_6(t)]^\top$ between the actual and the desired object configuration. The mathematical representation of prescribed performance for each element is given by the following inequalities:

$$-\rho_j(t) < e_j(t) < \rho_j(t), \; \forall t \geq 0, j \in \{1, ..., 6\} \tag{1}$$

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:

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

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 $\rho_{jo} > \left|e_j(0)\right|$. 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 only on 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 as well as velocity measurements and no explicit data is exchanged on-line between the robots, the object dynamics -which are considered known- along with the estimated acceleration are employed to compute the leader's applied wrench. As a leader could be consider either a human (Figure 7) or a robotic agent (Figure 6b). An abstraction of the proposed methodology is depicted in Figure 8 and Figure 9 for the SM and TG, respectively.

C. High-level Planning

Multi-agent task planning of heterogeneous robots is another part 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.

1) 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 formalised a catalogue of mission specification patterns containing recurrent specifications of missions[1]. Each pattern in the catalogue is formulated both in structured English and in LTL. Then, to enable the specification of more complex and sophisticated missions, we created a Domain Specific Language (DSL), called Promise [19]. Promise[2] uses the mission specification patterns as basic building blocks and enable the specification of a mission via the use fo composition operators. According to the SERA [10] architecture described above, a global mission specification is then 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 be then automatically translated into LTL formulations.

2) Temporal Logic Formulation: The Planning Module follows the ideas described in [20] and [21] and it consists of two main functionalities: firstly, 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 $\mathscr{M}_i, \forall i \in (1, 2, \ldots, N_R)$ as follows. The internal states of the robots, e.g. "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 $\mathscr{B}_i, \forall i \in (1, 2, \ldots, N_R)$ 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 $\mathscr{R}_i = \mathscr{M}_i \times \mathscr{B}_i$, which is the product of the motion and action tuples. More details can be found in [21].

In order to find a plan over $\mathscr{R}$ that satisfies the assigned task, we employ standard techniques from formal verification methodologies. First, the assigned task is expressed as a Linear Temporal Logic (LTL) formula φ, which is then converted to a Nondeterministic Büchi Automaton A_φ. Then, we construct the product $A_P = \mathscr{R} \times A_\varphi$ and by using graph search algorithms, we obtain a least-violating plan over $\mathscr{R}$. The plan is least-violating in the sense that it satisfies most of

[1]Mission specification patterns: http://roboticpatterns.com/

[2]Promise: https://promisedsl.wixsite.com/promise

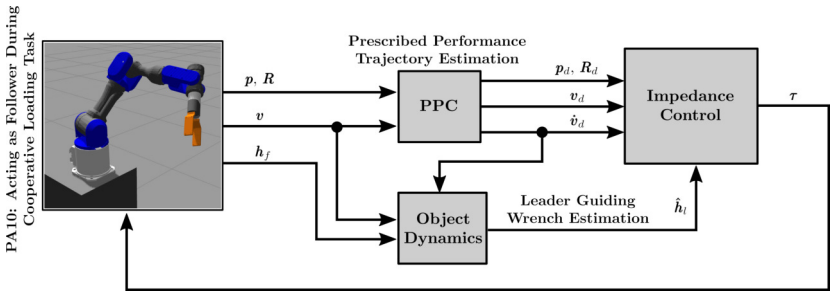


Fig. 8: Block Diagram of the Follower Robotic Agent's Control Scheme for the Desired Trajectory Estimation and Tracking.

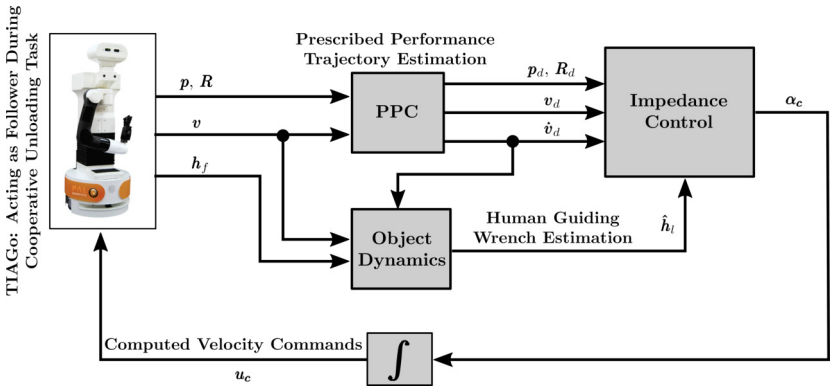


Fig. 9: Block Diagram of TG Control Scheme during Human-Robot Cooperative Transportation Procedure.

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 $\mathscr{R}$. A new plan is then computed to improve the satisfiability of the assigned task, given the new workspace information.

III. Application

A. Experimental Setup & Scenario

We propose a *single* scenario, which encapsulates all key concepts. More specifically, the scenario revolves around three (3) robotic entities: i) a mobile manipulator (*PAL TIAGo*), ii) a static manipulator (*Mitsubishi PA-10*), and iii) 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.

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 towards 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, also the human calls the mobile manipulator for help via an appropriate posture which is performed in front of the mobile platform. The human and the mobile manipulator grasp together 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. In case 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.

B. 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 activites. Following the objectives of the project, the three main criteria for evaluation were i) *Flexibility*, ii) *Robustness*, and iii) *Efficiency*. Objectives relating to these three goals are defined separately for each of the modules described in section II and the measures of success for these key objectives are specified in Table I.

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

	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 I: Measures of success for the proposed architecture.

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 datasets that provide ground truth pose measurements. The Table II 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* object) in order to realize the loading and unloading tasks. The ratios of successfully grasped objects that were recorded during the experiments are depicted in Table III. 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 6a depicts the initial pose of the static manipulator (SM) and the mobile platform (MP) 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 6b depicts the initial pose of the SM, the TG and the MP 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 TG'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 IV.

In Figure 11 and Figure 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 neither for the leader's trajectory nor for 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 seconds, for the plan derivation that satisfies the task is less than 0.1 seconds, and for plan synthesis and reconfiguration is also less than 0.1 seconds. 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 $\sim 10-30$ seconds. 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 the robots cannot execute an action they are modeled to. 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.

IV. Discussion

The proposed decentralized framework was tested in a scenario which included a sufficient number of tasks in order to assess the performance of each individual component, as well as the performance of the overall architecture. However, the multi-agent system consisted only of 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 set of high-level tasks. Moreover, an important issue that we have not yet addressed is the on–line re-configuration of planning in the presence of actuation or sensor faults, which also mandates for an increase in the number the involved agents. Additionally, a significant attribute that we intend to incorporate in the future is the one pertaining the aspects of cognitive safety in human-robot collaboration tasks. More specifically, we aim to tackle issues such as human-situation awareness, motion intention (inside the workspace), as well as inherited compliant behaviour during cooperative manipulation tasks. In this way, we will be able to assure that robots and humans can safely and effectively co-exist in a real dynamic environment.

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

TABLE II: Perception Module: Human and object detection/pose estimation evaluation.

	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.*

TABLE III: Ratio of Successful Object Grasping.

	Ratio of Success	Time Needed
PA-10 & SUMMIT	90% (9 successful out of 10 attempts)	38.5 *sec*
PA-10, TIAGo & SUMMIT	83% (10 successful out of 12 attempts)	48.2 *sec*

TABLE IV: Ratio and Time Needed of Successful Cooperative Loading Procedures.

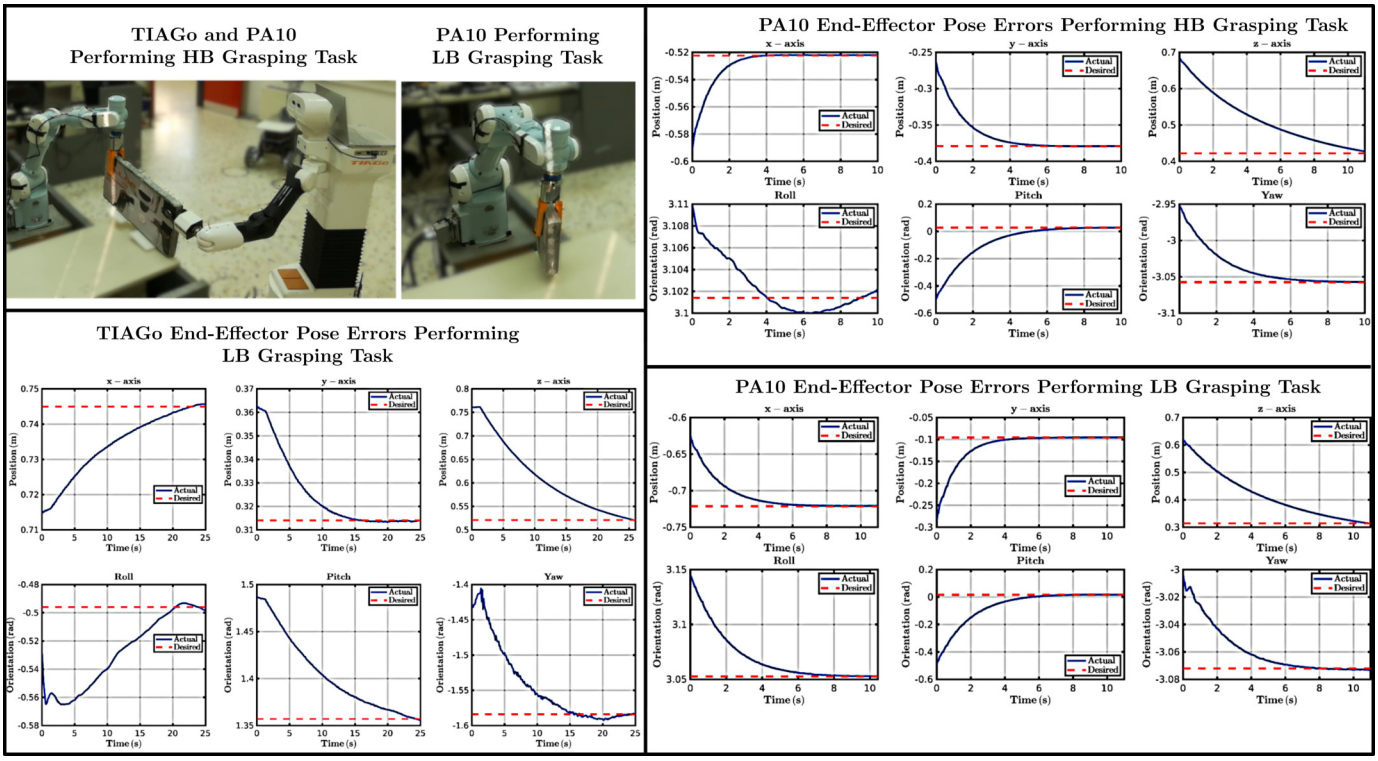


Fig. 10: Manipulators' End-Effector Errors Performing Grasping Tasks.

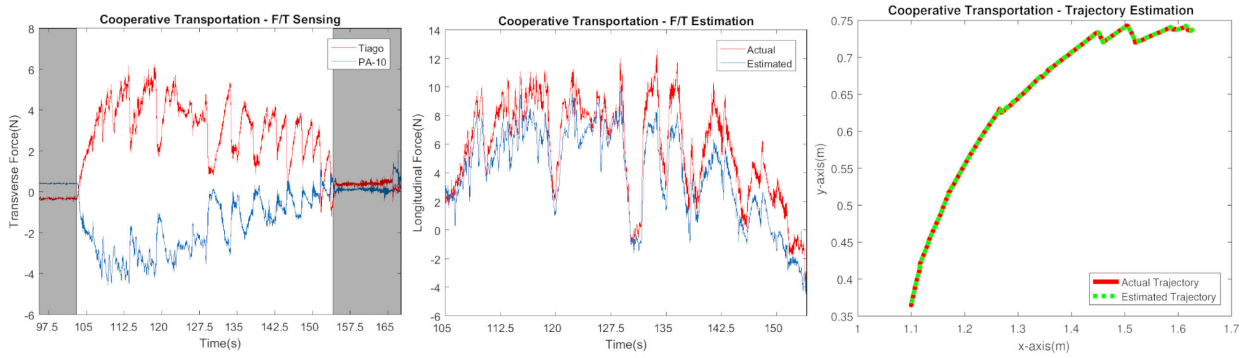


Fig. 11: Force/Torque Sensing-Estimation and Trajectory Estimation During Cooperative Loading Task: The left figure depicts the interaction forces during cooperative transportation task. The leader's (TG) interaction force as estimated by the follower (SM) and the actual one are depicted in the middle figure. The right figure shows the leader estimated 2D trajectory and the actual one.

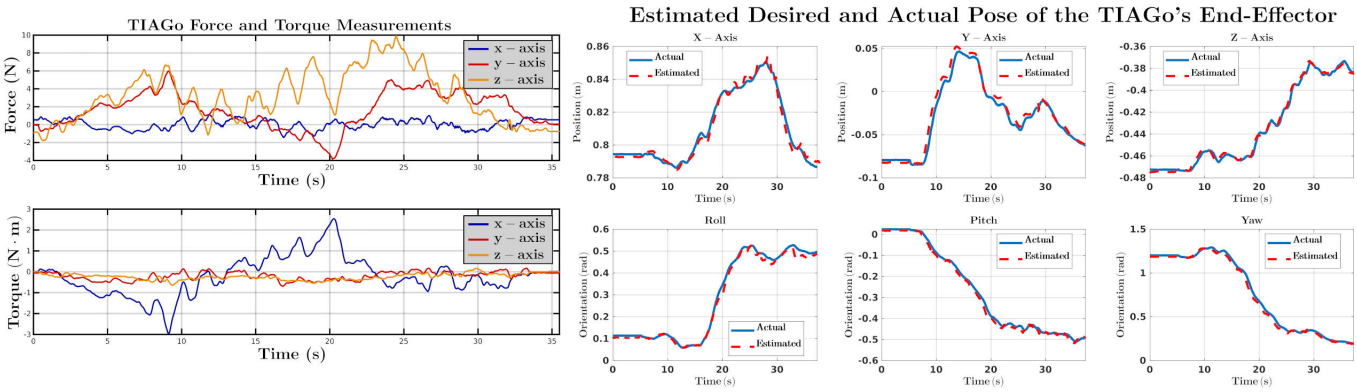


Fig. 12: Force/Torque Sensing and Trajectory Estimation in Cooperative Human-TIAGo Unloading Task

V. 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 semi-structured work-spaces. 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 multi-tasking 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.

Acknowledgement

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 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 *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, pp. 2923–2930, IEEE, 2014.

[2] S. M. LaValle, *Planning algorithms*. Cambridge university 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," *Autonomous Robots*, vol. 42, no. 4, pp. 801–824, 2018.

[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 *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 781–787, IEEE, 2017.

[5] Z. Liu, Y. Yamauchi, S. Kijima, and M. Yamashita, "Team assembling problem for asynchronous heterogeneous mobile robots," *Theoretical Computer Science*, vol. 721, pp. 27–41, 2018.

[6] H. Tanner, S. Loizou, and K. Kyriakopoulos, "Nonholonomic navigation and control of cooperating mobile manipulators," *IEEE Transactions on Robotics*, vol. 19, no. 1, pp. 53–64, 2003.

[7] W. C. Dickson, C. J. R. H., and S. M. Rock, "Decentralized object impedance controller for object/root-team systems: Theory and experiments," in *Proceedings of the IEEE ICRA*, pp. 3589–3596, 1997.

[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 *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, p. 17–22, 1997.

[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," *Robotics and Autonomous Systems*, vol. 103, pp. 122–133, 2018.

[10] S. García, C. Menghi, P. Pelliccione, T. Berger, and R. Wohlrab, "An architecture for decentralized, collaborative, and autonomous robots," in *IEEE International Conference on Software Architecture (ICSA) 2018*, 2018.

[11] M. Lourakis and X. Zabulis, "Model-Based Pose Estimation for Rigid Objects," pp. 83–92, Springer, Berlin, Heidelberg, 2013.

[12] A. Makris and A. A. Argyros, "Robust 3d human pose estimation guided by filtered subsets of body keypoints," in *Machine Vision Applications*, 2019.

[13] "Ros navigation stack." http://wiki.ros.org/navigation.

[14] P. Vlantis, C. Vrohidis, C. P. Bechlioulis, and K. J. Kyriakopoulos, "Robot navigation in complex workspaces using harmonic maps," in *IEEE/RSJ International Conference on Robotics and Automation (ICRA)*, IEEE, 2018.

[15] C. Rösmann, F. Hoffmann, and T. Bertram, "Integrated online trajectory planning and optimization in distinctive topologies," *Robotics and Autonomous Systems*, vol. 88, pp. 142–153, 2017.

[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 *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1–6, Oct 2018.

[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 *2019 IEEE/RSJ International Conference on Robotics and Automation (ICRA)*, IEEE, 2019.

[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 *2018 European Control Conference, Limassol, Cyprus*, vol. 1, pp. 249–254, 2018.

[19] S. García, P. Pelliccione, C. Menghi, T. Berger, and T. Bures, "High-level mission specification for multiple robots," in *Proceedings of the 12th ACM SIGPLAN International Conference on Software Language Engineering*, pp. 127–140, 2019.

[20] M. Guo and D. V. Dimarogonas, "Multi-agent plan reconfiguration under local ltl specifications," *The International Journal of Robotics Research*, vol. 34, no. 2, pp. 218–235, 2015.

[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 International Conference on Intelligent Robots and Systems (IROS)*, pp. 240–245, 2013.