

Pick-and-place in dynamic environments with a mobile dual-arm robot equipped with distributed distance sensors

Sotiris Stavridis¹, Pietro Falco² and Zoe Doulgeri¹

Abstract—Mobile bimanual manipulation in a dynamic and uncertain environment requires the continuous and fast adjustment of the robot motion for the satisfaction of the constraints imposed by the task, the robot itself and the environment. We formulate the pick-and-place task as a sequence of mobile manipulation tasks with a combination of relative, global and local targets. Distributed distance sensors on the robot are utilized to sense the surroundings and facilitate collision avoidance with dynamic and static obstacles. We propose an approach to kinematically control the robot by solving a priority constrained optimization problem online. Experimental results on the YuMi bimanual robot mounted on the Ridgeback mobile platform validate the performance of the proposed approach.

I. INTRODUCTION

Dual-arm manipulation and manipulators have been presented in detail in several review papers [1], [2]. The recent developments in the areas of humanoid robots and the work on learning by imitation has motivated the construction and use of dual-arm setups. As bimanual robots can mimic human behaviors, act and manipulate objects in ways similar to humans, human bimanual skills can be transferred to them. Consequently, the robot can work interchangeably and in collaboration with humans in the same task. Furthermore, mobile manipulation increases in popularity due to advancements in the control approaches and performance, the reduced costs and the need for mobility in robotic applications [3], [4]. Although, multiarm and mobile robots increase the complexity of the system, they offer multiple end-effectors to interact with the environment and the mobility to work on a broader workspace executing more complex tasks. Their intrinsic redundancy can be further exploited to satisfy constraints that ensure the safety of the robot and its environment, such as avoiding collisions and respecting its joint limits, as well as satisfying additional objectives. The majority of the works on dual-arm coordinated motion have focused on the cooperative task space representation via absolute and relative velocity variables ignoring environmental and self constraints that are present in such problems in practice [5], [6], [7].

Distance sensors provide a solution of low price and computational cost with fewer occlusion problems as opposed

to cameras for obstacle detection. The majority of research on collision avoidance with distance sensors focuses on repulsive motions using potential fields which force the robot to move in the opposite direction not taking advantage of all the feasible directions, e.g. [8]. In particular, in [8] external distance sensors are distributed on a mobile robotic arm to sense its surroundings and reactively avoid collisions by modifying pre-planned trajectories in the task space. Priority constraint based optimization is used in [9] for a robotic arm equipped with artificial skin consisting of multiple near-sighted proximity sensors with equality constraints for collision avoidance. In [10] inequality constraints are utilized; obstacles are sensed by distance sensors mounted on a robotic arm but no priority among the tasks and constraints is defined. Notice that the above works focus on single-arm robots.

In bimanual robots both collision avoidance with external obstacles and self-collision should be addressed. In [11] the danger field method is used for collision avoidance of a bimanual robot with external obstacles utilizing distance sensors. A task hierarchy is adopted where non-critical tasks are activated and deactivated depending on a scalar danger field metric, while evasive motions are projected into the null-space of the active tasks. Other works for bimanual robots with collision avoidance either assume known obstacle position or detect the obstacles visually. Self-collision avoidance and collision avoidance with known obstacles is addressed in [12] for a humanoid robot using a task priority strategy where each successive task is projected into the null space of the previous higher priority tasks. In our previous work [13], self and obstacle collision avoidance for a dual-arm robot is addressed together with relative motion tasks and performance metrics optimization utilizing a priority constrained optimization method.

Mobile platforms are predominantly used to navigate the environment and move the robot to a specific location followed by the execution of the manipulation task without further utilization of the mobile platform. However, with this strategy the mobile platform is not fully exploited since apart from mobility it provides additional degrees of freedom (dofs) that may be used to satisfy secondary objectives and constraints. Mobile manipulation involves the consideration of the dofs provided by the mobile platform and the arm as one kinematic chain. Mobile manipulation with single arm robots has been proposed for a service robot in [14] and for door opening in [15].

Mobile bimanual pick-and-place tasks are addressed in [16]. However, it relies on planning rather than real-time motion generation given the current visual input which is

The research leading to these results has received funding from the European Community's Framework Programme Horizon 2020 under grant agreement No 871704, project BACCHUS

This research was partially funded by ABB Corporate Research Center, Västerås, Sweden as part of the ABB Research Stay Initiative

¹Department of Electrical and Computer Engineering, Aristotle University of Thessaloniki, Greece, e-mail: {sotistav, doulgeri}@ece.auth.gr

²ABB Corporate Research, Department of Automation Solutions, Västerås, Sweden, e-mail: pietro.falco@se.abb.com

used to reconstruct the environment. It is claimed that the real-time nature of this representation allows the robot to react by replanning to unforeseen obstacles.

In this work we propose a mobile dual-arm robot equipped with distance sensors distributed on its body and a kinematic control method for a mobile bimanual pick-and-place task in a dynamic environment. We restrict our analysis in cases the manipulation target is near the robot, e.g. within the field of view of the robot, as a navigation solution is out of scope of this work. The pick-and-place task involves a sequence of tasks with local, global and relative targets. We formulate each task as a priority constrained optimization problem that is solved online to avoid collisions with dynamic and static obstacles and achieve the goal. The primary contribution of this work is an integrated framework for mobile bimanual robots with distributed proximity sensors applied in bimanual pick-and-place tasks in the presence of static and dynamic obstacles. Proximity sensing for obstacle avoidance enables high responsiveness for reactive collision avoidance, but existing works have not so far fully exploited their potential. This work focuses on pick-and-place tasks, however this framework can be generalized to other tasks that can be defined using the subtasks and constraints described in the following sections.

II. PROBLEM DESCRIPTION AND CONCEPT SOLUTION

A mobile bimanual manipulator is inherently a system of multiple kinematic chains and subchains. Thus, mobile bimanual tasks may involve a number of motion tasks that are more conveniently defined with respect to different frames of reference. These are the world frame $\{w\}$, the mobile platform frame $\{m\}$ and the left $\{l\}$ and right $\{r\}$ end-effector frames. By combining such motion tasks a robot is able to execute various complex tasks in many applications. Four types of motion tasks are distinguished on the basis of the reference frame used for their most convenient description:

- 1) The global end-effector motion task defined in $\{w\}$ which can involve the left, right or both end-effectors (6 or 12 dofs),
- 2) the local end-effector motion task defined with respect to $\{m\}$ which can involve the left, right or both end-effectors (6 or 12 dofs). Notice that in this case the target is invariant to the motion of the mobile platform,
- 3) the relative end-effector motion task (6 dofs) that is defined either in $\{l\}$ or $\{r\}$,
- 4) the mobile platform motion task (3 dofs) that is defined in $\{w\}$.

Additional tasks can be also defined on the basis of an optimization objective, such as velocity or force manipulability metrics for more efficient task execution. Examples of a global end-effector motion task is the unimanual or bimanual reaching of a pregrasp pose of an object located in the world. A local end-effector motion task is for example the placing of a grasped object on a tray that is located on the mobile platform. A relative end-effector motion task is the assembly of two small parts or the pose maintenance of a bimanually

grasped object during transfer. Finally, a mobile motion task is the motion of the mobile platform to reach near a place of interest, e.g. a desk or workbench.

Moreover, effective mobile bimanual task execution in a dynamic environment involve a number of constraints that should be respected. In this work three types of constraints are detailed, namely:

- 1) Joint limit avoidance, that involves position and velocity constraints,
- 2) self-collision avoidance, that involves collision avoidance of the arms with each other and the mobile platform
- 3) obstacle collision avoidance, that involves collision avoidance with static and dynamic obstacles, like humans moving in the workspace of the robot.

One can also list other types of constraints such as field of view, positioning of the center of mass, etc.

In this work a kinematic control approach with a hierarchical constrained optimization framework is adopted [17]. Robot tasks are defined by equality constraints which relate the desired velocity of the end-effector(s) to reach a global, local or relative target with the robot configuration space velocities via the forward kinematics of the appropriate kinematic chain. Inequality constraints are used to model structural or environmental constraints perceived by the sensors of the system, which should be respected during execution. The hierarchical optimization framework given equality and inequality constraints associated with an appropriate priority level, generates a configuration level velocity reference which is then commanded to the controllers of the robotic system. Thus, this control approach may be implemented on robots that offer position and/or velocity interfaces. Given all the equality and inequality constraints the robotic system cannot guarantee their satisfaction at all times, as conflicts between them may arise. The reference motion generated by a hierarchical optimization algorithm is optimal for the highest priority tasks, while lower priority tasks are satisfied only when they do not cause the violation of higher priority ones. Hence the definition of the appropriate priority between them influences the quality of the derived solution regarding the task objective.

In the hierarchical optimization framework the following hierarchical constrained optimization problem is solved at each control cycle to generate reference velocities for the robot dofs that optimally satisfy them.

$$\begin{aligned} \dot{q}^* &= \underset{\dot{q}, w_1, \dots, w_n}{lexmin} \{ \|w_1\| \dots \|w_n\| \} \\ \text{subject to } & A_k \dot{q} \leq b_k + w_k \quad \forall k = 1 : n \end{aligned} \quad (1)$$

where k denotes the index of a given priority level with 1 being the highest, n is the total number of hierarchical levels, matrix A_k and vector b_k describe the linear equality and inequality constraints, w_k are slack variables used to relax the constraints in case of non-feasibility, while *lexmin* denotes that the solution is minimal with respect to a lexicographic order of priority [17]. Lastly, \dot{q} is the velocity vector provided to the mobile dual-arm robot dofs. The position vector q may be decomposed to $q = [q_l^T \quad q_r^T \quad q_m^T]^T$, where q_l, q_r are the

joint position vectors of the left and right arm respectively, while q_m is the mobile platform dofs vector. Assuming an omnidirectional mobile platform moving on a planar surface its motion may be modelled as two prismatic and one revolute joints, thus $q_m \in \mathbb{R}^3$.

In this work we define the general framework for addressing the specific challenges related to mobile bimanual robot tasks in terms of the detailed design of tasks, constraints and hierarchies and test it in a complex bimanual pick-and-place operation. The implemented hierarchy is quite rich involving all the tasks and constraints defined in this section given the distributed sensing system of the robot. We thus demonstrate the effectiveness of the approach in a realistic non-trivial case. The following section details the formulation of equality and inequality constraints.

III. CONSTRAINT FORMULATION

A. Equality constraints

In this work a kinematically controlled robot is assumed. Let a reference trajectory in $SE(3)$ be described by the position p_d and orientation Q_d expressed as a unit quaternion, as well as their velocities v_d and ω_d respectively. A quaternion is given by $Q = [\eta \ \epsilon^T]^T$ with η being the scalar and ϵ the vector part of the quaternion. The relation between the quaternion rate \dot{Q}_d and the rotational velocity ω_d is given by: $\dot{Q}_d = \frac{1}{2} \begin{bmatrix} \eta & -\epsilon^T \\ \epsilon & \eta I_3 - \hat{\epsilon} \end{bmatrix} \begin{bmatrix} 0 \\ \omega_d \end{bmatrix}$ where $\hat{\epsilon}$ is the skew symmetric matrix of the quaternion vector part ϵ and I_3 is the identity matrix of size 3.

1) *Global end-effector motion task*: In the global motion task, the reference trajectories involve motion generation to a target pose for the end-effector(s) with respect to $\{w\}$. To reach the target pose, end-effector reference trajectories can be either planned or generated in real-time by a dynamical system, e.g. a Dynamic Movement Primitive (DMP) [18].

In the hierarchical framework we construct equality constraints for this task by providing the kinematic relationship between the task velocities v_{di}, ω_{di} with $i = \{l, r\}$ for the left and right end-effector and \dot{q} :

$$\begin{aligned} \begin{bmatrix} v_{dl} \\ \omega_{dl} \end{bmatrix} &= [J_l \ 0_r \ J_{ml}] \dot{q}, \text{ left arm} \\ \begin{bmatrix} v_{dr} \\ \omega_{dr} \end{bmatrix} &= [0_l \ J_r \ J_{mr}] \dot{q}, \text{ right arm} \end{aligned} \quad (2)$$

where $J_l, J_r \in \mathbb{R}^{6 \times 7}$ denote the respective individual arm Jacobians and $J_{mi} \in \mathbb{R}^{6 \times 3}$ with $i = \{l, r\}$ denotes the Jacobian which maps the mobile platform velocity to the end-effector velocity and $0_l, 0_r$ are zero matrices of the same dimensions as J_l and J_r respectively. Jacobian J_{mi} has zero values in all coordinates of $SE(3)$ that are not affected by the platform motion. In (2) the motion of an end-effector is generated by both the motion of the platform and the motion of the respective arm.

2) *Local end-effector motion task*: In case the task is defined in $\{m\}$, when for example the target is static with

respect to $\{m\}$ (for e.g. an object on the platform), it constitutes a local motion task and the respective Jacobian is:

$$\begin{aligned} \begin{bmatrix} {}^m v_{dl} \\ {}^m \omega_{dl} \end{bmatrix} &= [{}^m J_l \ 0_r \ 0_m] \dot{q}, \text{ left arm} \\ \begin{bmatrix} {}^m v_{dr} \\ {}^m \omega_{dr} \end{bmatrix} &= [0_l \ {}^m J_r \ 0_m] \dot{q}, \text{ right arm} \end{aligned} \quad (3)$$

where the upper left superscript denotes the frame of reference $\{m\}$ and 0_m is the zero matrix of the same dimensions as J_{mi} .

If a task involves only one end-effector, only one of the equality constraints in (2) or (3) is needed in this level. When both end-effectors are involved in this task type e.g. the reaching of a bimanual pre-grasping pose (12 dofs), we concatenate both the equality constraints of (2) or (3). Notice that the additional dofs of the platform generate a redundancy (total dofs ≥ 15) that facilitates the existence of a solution.

3) *Relative end-effector motion task*: In relative tasks the reference trajectory is expressed with respect to either the left $\{l\}$ or the right $\{r\}$ end-effector frame following [13]. In this work $\{r\}$ is selected. The equality constraint on the relative motion task level is formed by utilizing the relative Jacobian which connects the relative velocities ${}^r v_{dl}, {}^r \omega_{dl}$ with \dot{q} :

$$\begin{bmatrix} {}^r v_{dl} \\ {}^r \omega_{dl} \end{bmatrix} = \begin{bmatrix} {}^m R_r^T & 0_3 \\ 0_3 & {}^m R_r^T \end{bmatrix} [{}^m J_l \ -\Gamma_{rl} {}^m J_r \ 0_m] \dot{q} \quad (4)$$

where ${}^m J_l, {}^m J_r$ denote the respective individual arm Jacobians, $\Gamma_{rl} = \begin{bmatrix} I_3 & -\hat{p}_{rl} \\ 0_3 & I_3 \end{bmatrix}$ with $p_{rl} = {}^m p_l - {}^m p_r$ is used to compute the velocity of a point coincident with the origin of frame $\{l\}$ that is induced by the velocity of frame $\{r\}$, and 0_3 denotes the zero square matrix of size 3.

4) *Mobile platform motion task*: This task is used to move the mobile platform with respect to $\{w\}$. The equality constraint is formed by utilizing the forward kinematics of the mobile platform.

$$\begin{bmatrix} v_{dm} \\ \omega_{dm} \end{bmatrix} = [0_l \ 0_r \ J_m] \dot{q} \quad (5)$$

where v_{dm} are ω_{dm} are the linear and rotational velocities of the mobile platform with respect to $\{w\}$. Note that the rows of J_m, v_{dm} and ω_{dm} have non-zero values only for the two tangent and one normal directions to the planar surface of motion for the linear and rotational velocity respectively.

B. Self-collision Avoidance Constraints

For the purpose of self-collision avoidance a similar methodology as in our previous work [13] is followed to construct the constraints. The links and the body of the robot are modeled as capsules (Fig.1a). The method of Velocity Damper [19] is used to derive the general form of the constraint:

$$n^T [{}^m J_{li} \ -{}^m J_{rj} \ 0_m] \dot{q} \geq -\varepsilon_c (d_{ij} - d_s) \quad (6)$$

where ε_c is a positive damping parameter, d_s is a safety distance between the two moving capsules, $d_{ij} = \|p_i - p_j\|$ and $n_{ij} = (p_i - p_j) / \|p_i - p_j\|$ with p_1 and p_2 being the closest

points between the two capsules (Fig.1b) whose computation is analytically described in [20]. Matrices ${}^m J_{li}$, ${}^m J_{rj}$ are the translational Jacobians of left and right robot arm considering the beginning of the capsule as end-effector transformed to the point p_i and p_j respectively; these Jacobians are augmented with zero columns for the subsequent joints. In case a collision pair involves a body capsule, which is static with respect to $\{m\}$, the respective Jacobian is zeroed out. Each capsule pair

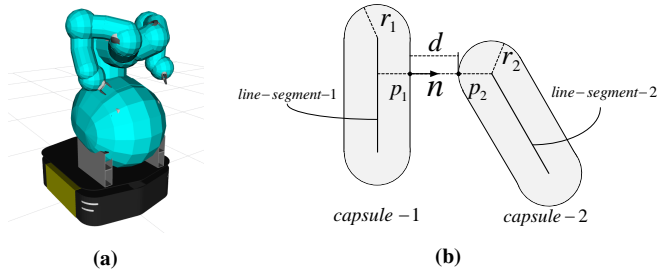


Fig. 1: a) The capsule-based collision model of the YuMi robot, b) a capsule-capsule collision pair.

constitutes a single inequality constraint. Collision pairs that cannot collide due to their geometry and structural limits of the robot are removed to reduce the number of constraints in the solver.

C. Obstacle Collision Avoidance constraints

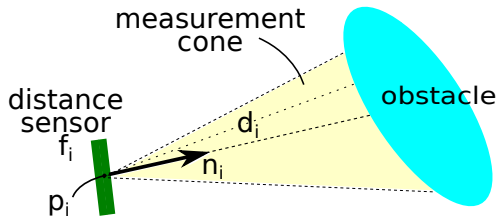


Fig. 2: Distance sensor view and measurement of an obstacle.

The utilization of distance sensors allows sensing of static and dynamic obstacles in the environment of the robot which in turn allows for environment-agnostic safe motion. Let f_i denote a given distance sensor attached to the robot with $i = 1 \dots F$ where F is the total number of distance sensors. Let the position and unit view direction of sensor f_i be $p_i(q)$ and $n_i(q)$ respectively, as seen in Fig.2. In this work, we use inequality constraints for sensor-based collision avoidance which are formulated as follows:

$$n_i^T [J_{li} \quad 0 \quad J_{mi}] \dot{q} \geq -\varepsilon_r (d_i - d_{si}), \quad \text{left arm} \quad (7)$$

$$n_i^T [0 \quad J_{ri} \quad J_{mi}] \dot{q} \geq -\varepsilon_r (d_i - d_{si}), \quad \text{right arm}$$

$$n_i^T [0 \quad 0 \quad J_{mi}] \dot{q} \geq -\varepsilon_r (d_i - d_{si}), \quad \text{platform} \quad (8)$$

with d_i being the distance measured, d_{si} a preset safety distance associated with f_i , ε_r is a positive damping parameter and J_{li} , J_{ri} , J_{mi} are the translational Jacobians mapping the left, right arm joint velocities and the mobile platform velocity respectively to the velocity of a point where sensor f_i is located. Notice that the measured distance d_i is the distance

from the sensor to the closest intersection point between the measurement cone of the sensor and the obstacle (Fig.2).

It is important to identify whether the sensors view an external obstacle or a part of the robot. If the measured distance corresponds to a point obstacle that lies inside a collision capsule of the robot, it is identified as a self-measurement and the corresponding constraint is deactivated.

D. Joint Limit Avoidance

Finally, joint limits are strict limits of the robotic hardware and must be respected at all costs. Joint limits usually include position and velocity limits. Their constraints can be implemented by damping the joint velocities near their mechanical limits as well as near their velocity limits via the following inequalities:

$$\max \begin{pmatrix} \varepsilon_l (q_{min} - q) \\ -V_q \end{pmatrix} \leq \dot{q} \leq \min \begin{pmatrix} \varepsilon_l (q_{max} - q) \\ V_q \end{pmatrix} \quad (9)$$

where ε_l are positive damping parameters, q_{min} , q_{max} are the lower and upper joint limits respectively, V_q is the vector of the absolute joint velocity limits, while \max and \min denote the element-wise maximum and minimum of the provided vectors respectively.

IV. MOBILE BIMANUAL PICK-AND-PLACE APPLICATION

The above framework has been applied in a mobile bimanual pick-and-place task utilizing the various tasks described in Sec. II. In particular, a mobile motion task was utilized in the initiation of the pick-and-place task to reach near the pick-up location and also to reach the place-down location. Global motion tasks were utilized to reach the pre-grasping poses and in combination with a relative motion task to grasp, to reach a pre-release pose and to release the object. A local in combination with a relative motion task was used to lift and retract the grasped object. Furthermore, to ensure the safety of the robot and the environment the constraints described in Sec. II were also utilized. Joint limit constraints, self-collision constraints and obstacle collision avoidance constraints along with the motion tasks were incorporated in a prioritized scheme in order to complete the pick-and-place task. Violation of joint limits not only generates infeasible motion, but can also damage the robot, while violation of the collision avoidance constraints might damage the robot and the environment. Thus, it is reasonable to adopt the following hierarchical order of constraints in descending order:

$$c_{jl} \succ c_{col} \succ c_{rel} \succ \begin{matrix} c_{glo} \\ c_{loc} \end{matrix} \succ c_{mob} \quad (10)$$

where c_{jl} , c_{col} refer to joint limits and collision avoidance that includes self and dynamic obstacle collision avoidance, while c_{rel} , c_{glo} , c_{loc} and c_{mob} refer to equality constraints for relative, global, local and mobile tasks respectively. Task c_{rel} has the highest priority among the equality constraints since its violation may cause drop or squeeze of the grasped object while the lowest priority equality constraint is c_{mob} since the mobile platform pose is a secondary objective in comparison to the pose of the end-effectors in this task.

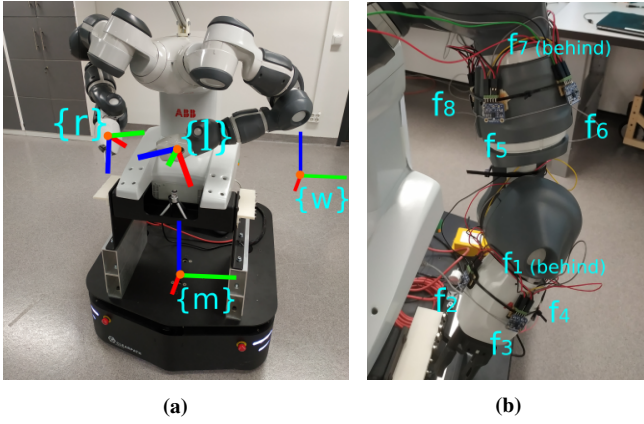


Fig. 3: a) The YuMi robot mounted on the Ridgeback mobile platform with the frames of reference displayed. b) The distance sensors placed on YuMi.

A. Experimental setup and implementation details

Our hardware setup consists of a YuMi bimanual robot mounted on a Ridgeback mobile platform (Fig.3a). Furthermore, 16 VL6180X distance sensors are placed on the arms of the robot (Fig.3b) taking measurements every $40ms$, as well as two Hokuyo UST-10LX lidar sensors are installed on the Ridgeback to perceive obstacles in the environment. Each of the utilized lidar sensors is offering 270° measurement angle with 0.25° resolution every $25ms$. Each such measurement constitutes a single obstacle collision avoidance constraint resulting in 2178 external collision avoidance constraints for all the utilized distance sensors. However only a small subset of them is active at every control cycle depending on the environment allowing for real-time performance.

The distribution of the sensors on the robot shown in Fig.3b was selected empirically, however a more optimized distribution may be computed through an optimization algorithm [21]. The calibration of the pose of the sensors with respect to the mobile frame $\{m\}$ was done manually through a graphical user interface. Furthermore, the AMCL ROS package was utilized for the localization of the mobile platform in a known map using lidar measurements and the odometry of the wheels. Finally, the constrained optimization problem formulated in the previous sections is solved using the HQP solver [17] as it is capable of computing a solution in real-time in similarly sized problems.

The reference trajectories for c_{glo} , c_{loc} , c_{rel} and c_{mob} are generated by utilizing a second order dynamical system both for translation and rotation given by:

$$\begin{aligned} \dot{v}_d &= \alpha(-v_d - \beta(p_d - p_T)) \\ \dot{\omega}_d &= \alpha(-\omega_d - \beta 2\eta_e \epsilon_e) \end{aligned} \quad (11)$$

where $\alpha = 5, \beta = 0.8$ are positive constant gains and p_T, Q_T the target position and orientation expressed as a unit quaternion. The system is integrated to provide a reference translational and rotational velocity $v_d, \omega_d \in \mathbb{R}^3$ and the reference position and orientation expressed as a unit quaternion p_d, Q_d . Variables η_e and ϵ_e in (11) denote the scalar and vector part of the quaternion error $Q_d * Q_T^{-1}$. During execution static and dynamic obstacles, tables and

human respectively, are present in the scene. Safety distance values d_s are chosen empirically to trade off the distance required to allow reaching the pick-and-place locations with avoidance of dynamic obstacles. For the sensors on the wrist the safety distance is $0.1m$, on the forearm $0.15m$ and on the mobile platform it ranges from $0.25m$ to $0.45m$ depending on their location. Damping parameters in (6), (7), (8) and (9) are set at values $\epsilon_r = 2, \epsilon_c = \epsilon_l = 0.5$, to allow a smooth reaction. The control cycle period is set to $8ms$ as a solution to the optimization could be reached under this time period. In cluttered scenes where the number of the active constraints increases, the solver may take more than $8ms$ to reach an optimal solution. In such cases the solver was interrupted prematurely at $8ms$ and the current suboptimal solution was considered without noticeable degradation of the quality of the solution.

B. Pick-and-place experiment

The goal of the experiment is to grasp bimanually an object from the pick-up location and release it on top of the container (place-down location) shown in the snapshots of Fig.4. The pick-and-place task can be segmented in multiple motion sequences that must be executed sequentially to successfully complete it. The target key-frames of each task may be provided by a higher level planning module or extracted by learning from demonstration [22]. In this experiment the object pose, pick-up and place-down location are considered known. The experiment is initiated with a mobile motion task in order to position the robot close to the pick-up location. At $t = 10s$ during the mobile motion a person walks near the robot and disturbs its task (Fig.4a). The robot senses the human with the distance sensors and avoids collision with him by modifying the mobile platform motion and by moving the robot arms away from collision. Two global motion tasks follow, one per end-effector, in order to reach a pregrasping pose. Then while the right end-effector global task is active, a relative motion task is executed to grasp the object. This relative motion task is maintained until the object release. Once the object is grasped, a local motion task is executed in order to lift and retract the grasped object, followed by a mobile motion task to reach close to the place-down location. At $t = 65s$ once again a person disturbs the robot which avoids him (Fig.4b). A global motion task is executed to reach a prerelease and a release pose and finally a relative motion task to release the object (Fig.4c).

The time evolution of the right end-effector with respect to $\{w\}$ and the relative pose between the two end-effectors are given by Fig.5 and Fig.6 respectively. The green and cyan vertical dotted lines denote the initiation of a task and the human disturbance respectively, while the dashed lines denote the commanded target pose. It is clear that the target key-frames are successfully reached. In Fig.7 the distance sensor measurements are displayed. Measurements from f_4, f_7 and f_8 are omitted since there were no obstacles in their view. Notice the distance sensors sensing the human at $t = 10s$ (left: $d_1 - d_3, d_5 - d_6$) and $t = 65s$ (left d_1, d_2, d_5, d_6 and right d_1). Then the robot quickly adjusts its motion and moves

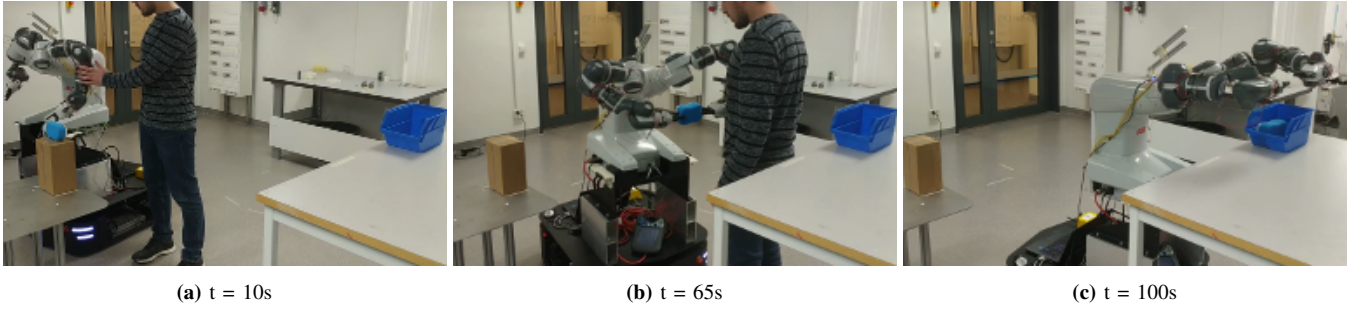


Fig. 4: Pick-and-place: Snapshots. See attached video.

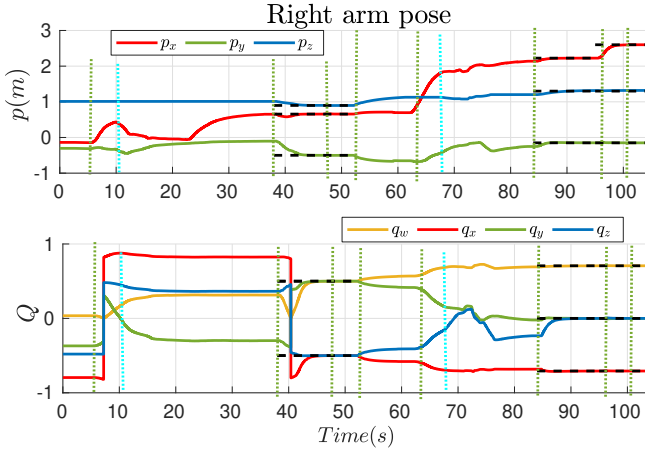


Fig. 5: Pick-and-place: Time evolution of the right arm end-effector pose with respect to the world frame.

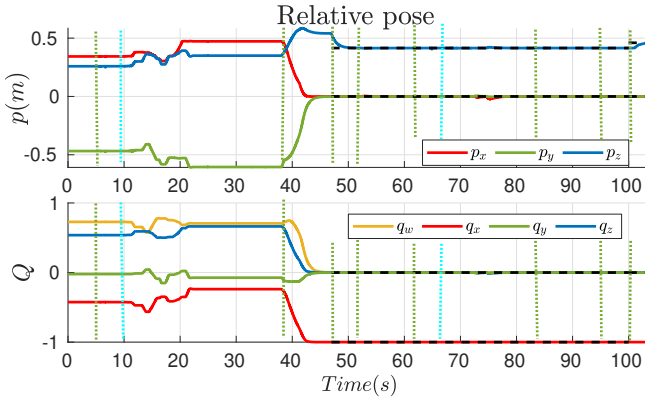


Fig. 6: Pick-and-place: Time evolution of the relative pose between the end-effectors.

away to avoid possible collisions. Notice that at $t = 65s$ in Fig.5 and Fig.6 the relative pose is maintained while the right arm pose is disturbed by the presence of the human. This happens due to the higher priority of the relative task compared to the right arm task. In comparison at $t = 10s$ where there is no active relative or end-effector task, the relative pose Fig.6 is visibly disturbed to avoid collision. Slight violations occur momentarily due to low measurement and control frequency with respect to human speed. After $85s$ the table is sensed (left: d_1 and right: d_2, d_3). The rest of the measurements correspond to self-measurements.

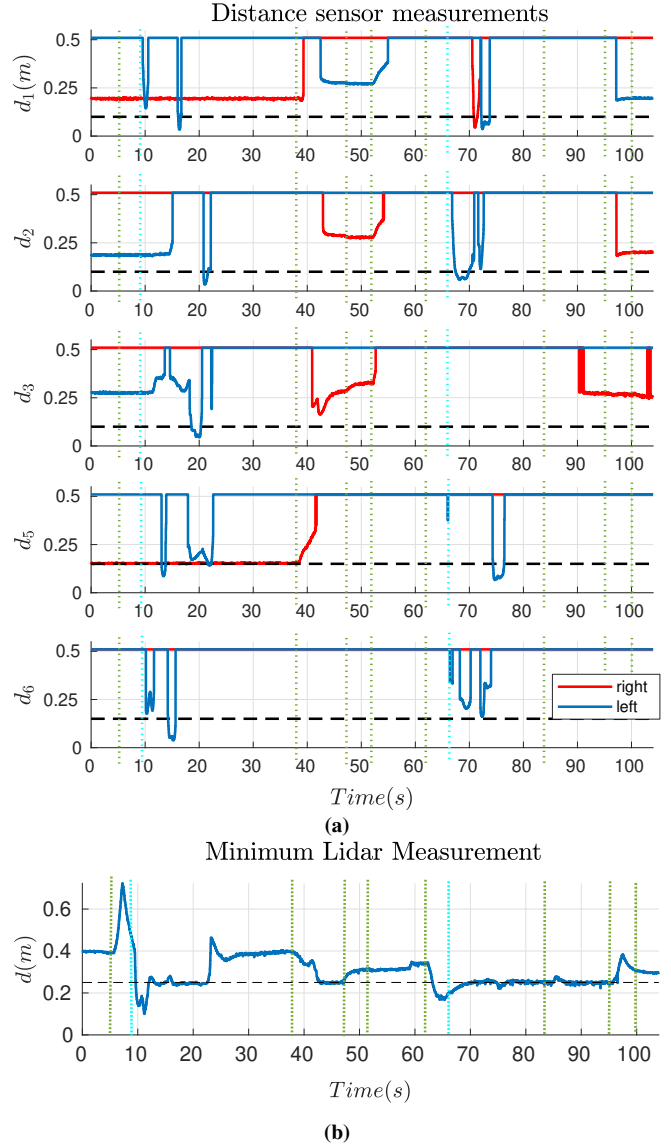


Fig. 7: Pick-and-place: Distance sensors measurements saturated at $0.5m$ for left and right arm (a) and lidar measurements (b). Dashed lines denote the safety distance.

C. Pose maintenance experiment

In order to further demonstrate the efficacy of the proposed method in case of demanding tasks such as maintaining the pre-release pose, an additional experiment was performed. In particular, the robot is commanded to maintain the right end-effector and the relative pose at their initial values.

The human is purposely approaching the mobile platform multiple times (at 7, 10, 12, 20, 22 and 25s) as seen in Fig.8. The robot is avoiding the person while the arms are adjusting their configuration in order to keep the relative pose and right end-effector pose unaffected. The relative pose is maintained with position errors less than $10^{-3}m$ and orientation errors less than $5 \cdot 10^{-3}rad$. The right end-effector orientation is maintained with error less than $5 \cdot 10^{-3}rad$, but the position error may reach up to $2.5 \cdot 10^{-2}m$ indicating considerable constraint violations. This can be explained by the lower priority of the right end-effector task with respect to the relative task. Nevertheless, it converges to the desired target as soon as the higher priority constraints allow it. Also, small violations of the distance sensor constraints occur only momentarily for the same reasons discussed in the previous experiment.

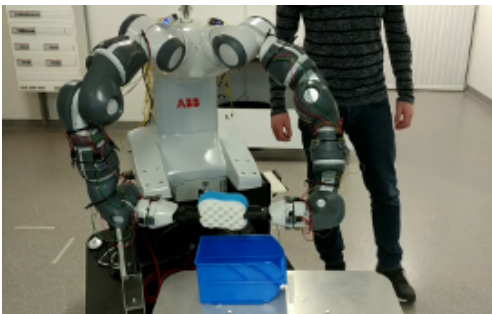


Fig. 8: Pose maintenance at $t = 15s$. See attached video.

V. CONCLUSIONS

In this work we formulate the pick-and-place task as a sequence of mobile manipulation tasks with relative, global and local targets in each task. Constraints for avoidance of joint limits, self-collisions and dynamic obstacles as well as equality constraints for task goals are defined in a hierarchical order. A hierarchical constrained optimization problem is then solved at each control cycle to generate reference velocities for the robot dofs that optimally satisfy them. The robot perception of the environment is robot centric with multiple distance sensors placed on the robot. Experiments with the YuMi robot mounted on the Ridgeback mobile platform demonstrate the efficacy of the proposed method even in case of demanding tasks. The developed methodology is modular thus allowing our framework to be easily adapted to new mobile bimanual robotic tasks.

REFERENCES

- [1] R. C. Dorf, S. Y. Nof, Concise International Encyclopedia of Robotics: Applications and Automation, John Wiley & Sons, Inc., 1990.
- [2] C. Smith, Y. Karayiannidis, L. Nalpantidis, X. Gratal, P. Qi, D. V. Dimarogonas, D. Kragic, Dual arm manipulation—a survey, *Robotics and Autonomous Systems* 60 (10) (2012) 1340 – 1353. doi:https://doi.org/10.1016/j.robot.2012.07.005.
- [3] M. Shneier, R. Bostelman, Literature review of mobile robots for manufacturing, National Institute of Standards and Technology, US Department of Commerce (2015).
- [4] R. Bostelman, T. Hong, J. Marvel, Survey of research for performance measurement of mobile manipulators, *Journal of Research of the National Institute of Standards and Technology* 121 (2016) 342–366.
- [5] B. V. Adorno, P. Fraisse, S. Druon, Dual position control strategies using the cooperative dual task-space framework, in: 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2010, pp. 3955–3960. doi:10.1109/IROS.2010.5650218.
- [6] H. A. Park, C. S. G. Lee, Extended cooperative task space for manipulation tasks of humanoid robots, in: 2015 IEEE International Conference on Robotics and Automation (ICRA), 2015, pp. 6088–6093. doi:10.1109/ICRA.2015.7140053.
- [7] D. Almeida, Y. Karayiannidis, Cooperative manipulation and identification of a 2-dof articulated object by a dual-arm robot, 2018, pp. 1–5. doi:10.1109/ICRA.2018.8460511.
- [8] P. Falco, C. Natale, Low-level flexible planning for mobile manipulators: a distributed perception approach, *Advanced Robotics* 28 (21) (2014) 1431–1444.
- [9] C. Vergara, G. Borghesan, E. Aertbelien, J. De Schutter, Incorporating artificial skin signals in the constraint-based reactive control of human-robot collaborative manipulation tasks, in: 2018 3rd International Conference on Advanced Robotics and Mechatronics (ICARM), 2018, pp. 280–287. doi:10.1109/ICARM.2018.8610697.
- [10] Y. Ding, U. Thomas, Collision avoidance with proximity servoing for redundant serial robot manipulators, in: 2020 IEEE International Conference on Robotics and Automation (ICRA), 2020, pp. 10249–10255. doi:10.1109/ICRA40945.2020.9196759.
- [11] N. M. Ceriani, A. M. Zanchettin, P. Rocco, A. Stolt, A. Robertsson, Reactive task adaptation based on hierarchical constraints classification for safe industrial robots, *IEEE/ASME Transactions on Mechatronics* 20 (6) (2015) 2935–2949. doi:10.1109/TMECH.2015.2415462.
- [12] O. Stasse, A. Escande, N. Mansard, S. Miossec, P. Evrard, A. Kheddar, Real-time (self)-collision avoidance task on a hrp-2 humanoid robot, in: 2008 IEEE International Conference on Robotics and Automation, 2008, pp. 3200–3205. doi:10.1109/ROBOT.2008.4543698.
- [13] S. Stavridis, Z. Doulgeri, Bimanual assembly of two parts with relative motion generation and task related optimization, in: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2018, pp. 7131–7136. doi:10.1109/IROS.2018.8593928.
- [14] E. Brzozowska, O. Lima, R. Ventura, A generic optimization based cartesian controller for robotic mobile manipulation, in: 2019 International Conference on Robotics and Automation (ICRA), 2019, pp. 2054–2060. doi:10.1109/ICRA.2019.8793635.
- [15] A. Jain, C. C. Kemp, Pulling open doors and drawers: Coordinating an omni-directional base and a compliant arm with equilibrium point control, in: 2010 IEEE International Conference on Robotics and Automation, 2010, pp. 1807–1814. doi:10.1109/ROBOT.2010.5509445.
- [16] S. Chitta, E. G. Jones, M. Ciocarlie, K. Hsiao, Mobile manipulation in unstructured environments: Perception, planning, and execution, *IEEE Robotics Automation Magazine* 19 (2) (2012) 58–71. doi:10.1109/MRA.2012.2191995.
- [17] A. Escande, N. Mansard, P.-B. Wieber, Hierarchical quadratic programming: Fast online humanoid-robot motion generation, *The International Journal of Robotics Research* 33 (7) (2014) 1006–1028. doi:10.1177/0278364914521306.
- [18] A. J. Ijspeert, J. Nakanishi, H. Hoffmann, P. Pastor, S. Schaal, Dynamical movement primitives: Learning attractor models for motor behaviors, *Neural Computation* 25 (2) (2013) 328–373. doi:10.1162/NECO.a.00393.
- [19] B. Faverjon, P. Tournassoud, A local based approach for path planning of manipulators with a high number of degrees of freedom, in: Proceedings. 1987 IEEE International Conference on Robotics and Automation, Vol. 4, 1987, pp. 1152–1159. doi:10.1109/ROBOT.1987.1087982.
- [20] A. D. Santis, A. Albu-Schaffer, C. Ott, B. Siciliano, G. Hirzinger, The skeleton algorithm for self-collision avoidance of a humanoid manipulator, in: 2007 IEEE/ASME international conference on advanced intelligent mechatronics, 2007, pp. 1–6. doi:10.1109/AIM.2007.4412606.
- [21] N. M. Ceriani, G. B. Avanzini, A. M. Zanchettin, L. Bascetta, P. Rocco, Optimal placement of spots in distributed proximity sensors for safe human-robot interaction, in: 2013 IEEE International Conference on Robotics and Automation, 2013, pp. 5858–5863. doi:10.1109/ICRA.2013.6631420.
- [22] G. S. Piperagkas, I. Mariolis, D. Ioannidis, D. Tzovaras, Key-frame extraction with semantic graphs in assembly processes, *IEEE Robotics and Automation Letters* 2 (3) (2017) 1264–1271. doi:10.1109/LRA.2017.2662064.