Available online at www.sciencedirect.com

ScienceDirect

Procedia Manufacturing 51 (2020) 454–461

Procedia MANUFACTURING

www.elsevier.com/locate/procedia

30th International Conference on Flexible Automation and Intelligent Manufacturing (FAIM2020)
15-18 June 2020, Athens, Greece.

A Safe and Energy Efficient Robotic System for Industrial Automatic Tests on Domestic Appliances: Problem Statement and Proof of Concept

Wendwosen Bellete Bedada[a,*], Rawan Kalawoun[a], Ismayil Ahmadli[a], Gianluca Palli[a]

[a]*DEI - Department of Electrical, Electronic and Information Engineering - University of Bologna, Viale Risorgimento 2, 40136 Bologna, Italy*

Abstract

In this paper, the design and the development of a robotic platform conceived to perform accelerated life tests on a newly manufactured domestic appliances is presented. The proposed system aims at improving the safety of human operators that share the workspace with the robotic platform which is a common scenario of test laboratories. A deep learning algorithm is used for the human detection and pose estimation, while the integration between a conventional motion planning algorithm with a fast 3D collision checker has been implemented as a global planner plugin for the ROS navigation stack. With the twofold objective of improving safety and saving energy in the battery-powered mobile manipulator used in this project, the problem of minimizing the overall kinetic energy is addressed through a properly designed task priority controller, in which the manipulator inertia matrix is used to weight the joint speeds while satisfying multiple robotic tasks according to a hierarchy designed to interact with the appliances while preserving the safety of the human operators. Simulations are carried out to evaluate the overall control architecture and preliminary results indicate the effectiveness of the developed system in the test laboratory floors.

Keywords: Autonomous mobile robots, Safety, Energy efficiency, Task priority control, Deep learning, Automatic testing

1. INTRODUCTION

Robots, nowadays are widely exploited to perform tasks commonly found in the manufacturing sectors that are fast, dangerous or repetitive for humans. In this regard, home appliance manufacturing industry has shown a strong demand for rapid and efficient ways to test new products to meet the growing customers demand for high performance appliances and compete in the market. The application of mobile manipulators in domestic appliance test facility is useful to relieve human operators from time consuming tasks, speed up the tests and improve efficiency.

However, alleviating the concern for safety in an environment shared by human operator and a robot plays essential part in the development of automated systems. Accelerated Life Test (ALT) is a commonly used technique to acquire reliability information quickly [3]. In particular, appliance manufacturing industry conducts ALT on the small fraction of appliances statistically sampled from the production line and are supervised by a human operators. The industrial robot safety standard given in [18] outlines safety requirements in a human-robot collaborative workspace. Speed and separation monitoring criterion aims at preventing physical impact from occurring by maintaining a static safe separation distance between the robot and any humans walking through the collaborative workspace. The safeguards for speed and separation monitoring criterion in a collaborative scenario includes, among other thing, a human detection system and a reduced speed mode for the robot depending on the separation distance when human operator is detected [10]. The development of robotic systems that fulfills this safety requirement involves integration of complex software and hardware subsystems in a reliable way. Even more, it is necessary to tailor the development to fit in to the particular work environment where humans and the robot collaborate.

Introducing robotic technology to the test and verification labs in appliance industry, while offering numerous advantages, it also poses new challenges with respect to safety. To meet the growing demand from customer side, the need to test products in short time and bring to the market is rising. Applying robots here can help in speeding up the test running the tasks non-stop in night shifts and weekends. It is also reasonable to consider mobile manipulators for these test labs to fully take advantage

[*] Corresponding author.

10.1016/j.promfg.2020.10.064

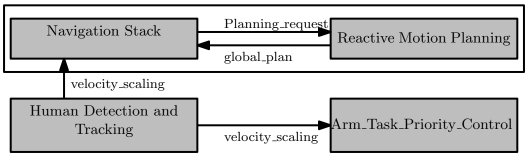


Fig. 1: Proposed safety system.

of the above mentioned points due to the extended workspace they over while also minimizing the cost associated to the test as sensors and the end-effector are mounted on a mobile agent and can be shared for all appliances [2, 4]. However, the safety requirement for mobile manipulators should be defined uniquely to ensure safety with respect to the mobile platform as well as the robotic arm. Defining the safety requirement could be even more complex if the simultaneous motion of the mobile base and the arm is considered.

This work presents safety requirements associated with human-robot collaboration in a domestic appliance test lab and proposes a safety oriented robotic system with energy efficient arm controller. This work utilizes off-the-shelf (OTS) mobile manipulators, the TIAGO robot from PAL Robotics, equipped with redundant arm and RGB-D camera to be deployed in the life test laboratory of washing machines. A deep learning approach is adopted to detect the humans in the scene, providing to the robot controller their pose estimation. Moreover, a global planner plugin for the ROS navigation stack has been implemented through the integration between a conventional motion planning algorithm and a Graphic Processing Unit (GPU) based fast 3D collision checker. To improve both the safety of the system and to save energy in the battery-powered mobile manipulator, the overall kinetic energy of the robot is minimized by the task priority controller through the use of the manipulator inertia matrix to weight the joint speeds. The task priority controller ensures also to satisfy multiple robotic tasks to perform proper interaction with the appliances while preserving the safety of the human operators. Simulations and preliminary experiments have been carried out to evaluate the system capabilities and develop all the required functionalities, and the results are reported in this paper.

The rest of the paper is organized as follow: In Section 2 the washing machine life test lab setup and the safety criterion for ensuring safety are discussed. The overall robotic system and the software and hardware requirements are also highlighted in this section. Section 3 discusses a deep-learning based approach for human operators detection and recognition in the workspace. A customized navigation system with the capability to plan motion in real-time with fast 3-D collision checking using range sensor data will be presented in section 4. Section 5 discusses a task priority control framework that minimizes energy consumption of the robotic system. Section 6 presents the simulations and experimental results to demonstrate the results achieved. Finally, in Section 7 concluding remarks and directions for future improvements are discussed.

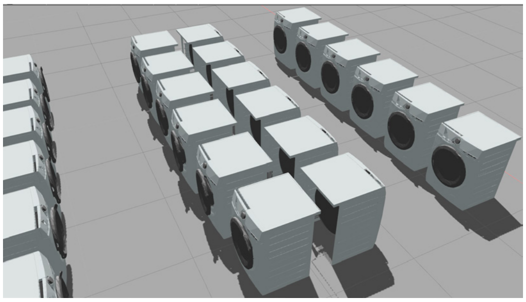


Fig. 2: Test Lab. layout in Gazebo Simulation Environment

2. System Description and Safety Requirement

Apart from detecting obstacles, the robotic system should be able to distinguish the presence of human operators in the environment. This allows the robot the enforcement of a safe velocity during the workspace sharing with the human operators. The overall system structure shown in Fig. 1 considers safety in a distributed manner in such a way that both the robotic arm and the mobile base fulfills these requirements.

The navigation system must achieve a 3D collision checking during the motion planning for navigation in a partially structured indoor environment and manage unpredictable changes due to the presence of human operators and unknown obstacles. Moreover, the navigation system should also be able to predict future collisions and demonstrate reactive motion planning capabilities. After reaching the goal, the system should be capable of detecting the pose of the washing machine as a whole and the different parts of the appliance such as knobs, buttons and door handle in particular. Finally, the system should also manipulate and physically interact with the products while avoiding collision and unsafe maneuvers. The list of tests that the robot has to set or perform on the appliance must be loaded to the robot in real-time through a wireless interface from external system. Depending on the size of the product (small, medium and large), the system should flexibly reconfigure itself to interact with each appliance.

For the purpose of testing the proposed system, a TIAGO mobile manipulator [13] has been used. However, different commercial robotic solution will be proposed in the future for deployment of the developed application in the appliance test lab. TIAGO is completely based on the Robot Operating System (ROS), that facilitates access to hardware and hides the complexities of transferring data between components [16].

The test lab contains large number of appliances that are arranged in a row with pair of rows facing each other and separated by a gap that allow appliance transportation as indicated in Fig. 2. This should be considered as an additional constraint for motion planning of the arm and the base of the robot.

3. Human Detection and Pose Estimation

Almost all robotic systems relies on a robust perception system to demonstrate autonomous and intelligent behaviors. Per-

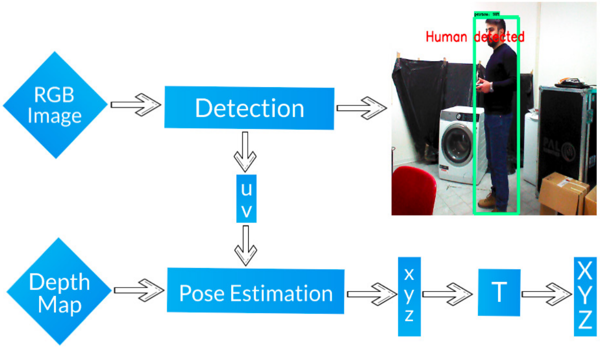


Fig. 3: Human Detection and Pose Estimation Procedure.

ception algorithms have been dominated mainly by computer vision algorithms that involves usage of descriptive features for object detection and recognition which usually involved human operator in the past. The emergence of Deep Learning (DL), however, introduced the concept of end-to-end learning where feature selection and extraction as well as classification is performed in one big learning network [12].

We have applied human detection and estimation of the human pose in order to provide the required safety and to eliminate the risk of any damage to human body in the working space of the robot. To implement human detection, Single Shot Detector (SSD300) [9], which has been adopted to MobileNetV2 [17] neural network, was used in this step. The neural architecture has previously been trained on COCO dataset [8].

The human pose with respect to the robot mobile base must be estimated for safety reasons. After detection step, the corresponding points on the image frame $[u,\ v]^T$ given by detection algorithm, are used as input to the pose estimation step, which produces $[x,\ y,\ z]^T$ coordinates of the corresponding points with respect to the camera optical frame as output. For that goal, the depth map of the scene is obtained using the RGB-D camera of the robot. Data encoded on the depth map represents the distance of the arbitrary point with respect to camera optical frame. After human detection and pose estimation in the robot workspace, the estimated values of the pose are used to implement safety measures for a shared worskpace.

4. Navigation Module

Collision detection and path monitoring consumes up to 90 percent of computation time during robot motion planning task [15]. This implies that fast collision detection plays central role in enforcing reactive behavior that interleaves planning, continuous motion validation and execution. The advent of fast computing capability has improved the development fast collision checking and re-planning algorithms that allows to realize robust navigation system for mobile robots. Computations related to collision detection and proximity queries on GPU can be more advantageous because of their multi-thread capability. A comparison of mesh based collision checkers with CPU implementation [14] and voxel based approach on GPU which discretize the robot and the environment is given in [6]. The voxel based implementation of collision detection on GPUs has shown improved results compared to the mesh based CPU implementation in the case of multiple queries of collision checking.

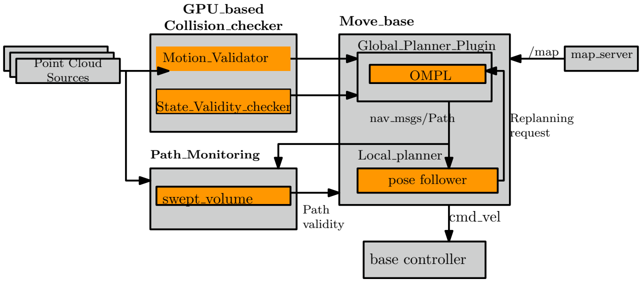
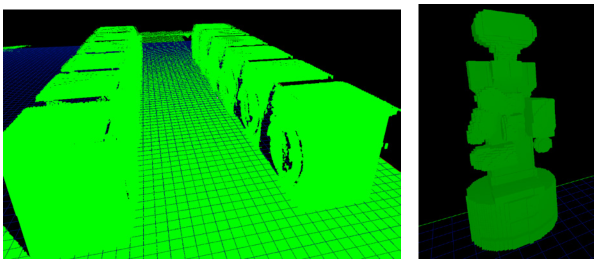


Fig. 4: GPU-Based Navigation Stack.

In this work, we propose a new approach that utilizes GPU-based collision checker given in [5] for motion planning in the framework of ROS navigation stack. A sampling based motion planner and a GPU-based state validity checking implementation forms the global planner plugin. A sampling based motion planning implementation in Open Motion Planning Library (OMPL) [20] is utilized as the global planner. The main difference between the proposed approach and the standard navigation stack is that we abandoned the use of 2D-costmap for environment representation. Instead, a 3D voxel-map created from depth sensors is utilized to facilitate a more reliable global planning that considers a three dimensional scene.

A local planner that follows the global plan is combined with an independent path monitoring implementation utilizing a swept volume of the robot used to detect future collisions and leveraging on the computational capabilities of modern GPUs. The architecture of the proposed GPU-Based navigation algorithm is reported in Fig. 4. During the execution of a plan, the collision checker monitors the global plan until the goal is reached. This is achieved by creating a voxelized swept volume of the robot along the planned path and detecting colliding voxels in the environment map. The swept volume is generated by keeping the results of path planning in the Voxel-Map to create a virtual corridor for the robot. The generation of the voxelized representation of the robot is performed offline and inserted into the map. The core idea of sampling based motion planning (SBMP) is the approximation of the connectivity of the sampled search space with a graph structure. The variations in the different types of SBMP originates from how the sampling of the planning space is performed to create the vertices of the graph and the strategies employed to generate the edges from it. The sampled states and connection between them forms the vertices and edges of the graph respectively. It obvious, however that the graph should constructed only from the valid vertices and edges which denote the valid states and path segments connecting them. Validating the sampled states and the connection among them is where the fast collision checking comes into play.



(a) Environment map in GPU voxels visualizer.

(b) TIAGO robot voxel representation.

Fig. 5: Voxel-based representation of the robot and working environment.

OMPL facilitate the integration with external collision checking algorithms through two abstract classes named *StateValidityChecker* and *MotionValidator*. The former allows the planner to evaluate the validity of states while the later validates motions between two specified states, effectively creating valid vertices and segments.

The *StateValidityChecker* class implements a routine that takes sampled state space and return the validity information. The path segment between the valid portion of the sampled state space is also validated to form the edges through a routine in the *MotionValidator* class of OMPL. The GPU-based collision checking is used to implement state validity checker for OMPL using a swept volume technique given in [6] in which the virtual model of the robot in Fig. 5(b) is inserted into the robot map at every pose we are checking for validity. The validity of the states is verified by comparing it to the corresponding occupancy of the voxels in the environment map. If the same voxel location in the robot and environment map is occupied, the particular state associated with the robot pose carrying the colliding voxels will be invalid. In case of motion validity checker, After valid states are sampled, a linearly interpolated path segment between two valid state creates the swept volume by inserting the discrete model of the robot along this path segment.

The environment map carries a discrete representation about the occupancy of the scene in a probabilistic way. To create this environment map, a transformed and filtered pointcloud from the robot's onboard RGB-D camera is inserted and updated on the GPU memory for fast collision detection. To perform the transformation of the pointcloud from camera frame of the robot to world frame, an Adaptive Monte Carlo Localization (AMCL) algorithm is used to localize the robot thereby transforming the points to the world frame in real-time. The environment map shown in Fig. 5(a) contains voxelized representation of two rows of washing machine partially seen from the TIAGO on-board camera.

As highlighted above, collision checking and monitoring are performed virtually using the environment scene and the voxelized robot model. The mesh model of the robot is rasterized into a binary 3D voxel grid to create a binary voxel model of the robot using an offline software given in [11]. This offline generated voxel representation of the robot will be inserted into a separate robot map according to the joint state and odometry information of the actual robot. By inserting the voxelized shape for each corresponding mesh surface on the robot, a full discrete representation of the robot is added into a robot map with same dimension as the environment map. The resolution of discretization of the mesh surfaces of the robot affect the memory usage and collision checking performance and should be selected carefully. The voxelized shape of the TIAGO robot given in Fig. 5(b) is used throughout this paper.

The motion planning developed in this work takes into account the presence of human to enable the robot to operate differently from the basic obstacle avoidance problem to ensure the safety of humans and enforce collaborative behaviour during the interaction. To this end, a virtual cylindrical volume is inserted in to the map in real-time to allow the motion planner to plan safe path. During execution, the velocity of the robot is varied depending on the distance between the robot and the detected human operator in the environment as 1) Full Operational, 2) Human-Aware and 3) Danger mode. In the scenario where either human is not in the worskpace or the distance between the robot and human operator is greater than certain predefined threshold, the robot operates in the Full Operational mode exploiting its full velocity as generated by the local planner. In the Human-Aware operation mode, however, the robot is at a distance that can injure the human operator, if they come in contact. Thus, the robot operates in a reduced velocity mode proportional to the distance. Finally, the Danger mode sets both linear and rotational velocity to zero because of the overlapping of the human safety zone and the robot's footprint.

The robot safety features are implemented at both the global and local planner level inside the software architecture of the GPU-Based navigation stack. At the global planner level, the presence of human operator is accommodated by inserting and updating a virtual safety zone during motion planning. The planner considers a cylindrical volume larger than the footprint of the human operator as a safety region. This imposes additional planning constraints that insure no part of the 3D robot shape violates the safety zone around the human. The local planner is responsible to generate desired velocity commands to the robot base and thus, has a direct influences on how the robot executes the global plan. Therefore, safety related behaviours of the robot are implemented in the local planner as follow:

$$\begin{cases} \text{Full Operational Mode} & \text{if} \quad \left\|x - x_{\text{safety}}\right\| > D_{\text{safe}} \\ \text{Human-Aware Mode} & \text{if} \quad D_{\text{danger}} < \left\|x - x_{\text{safety}}\right\| < D_{\text{safe}} \\ \text{Danger Mode} & \text{if} \quad \left\|x - x_{\text{safety}}\right\| < D_{\text{danger}} \end{cases} \tag{4.0.1}$$

where the x is the mobile robot's pose, x_{safety} is the center of the virtual safety volume around the human, D_{safety} is the minimum distance between the robot and the human before the local planner switches to the Human-Aware mode and D_{danger} the minimum safe distance between the human and robot.

5. Task Priority Control Framework

Task priority control enable the execution of several robotic tasks, such as singularity avoidance, joint limit avoidance, col-

.

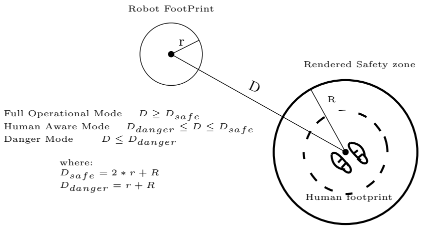

Robot FootPrint
r
D
Rendered Safety zone
R
Full Operational Mode $D \geq D_{safe}$
Human Aware Mode $D_{danger} \leq D \leq D_{safe}$
Danger Mode $D \leq D_{danger}$
where:
$D_{safe} = 2 * r + R$
$D_{danger} = r + R$
Human footprint

Fig. 6: Safety requirement description.

lision avoidance, joint speed limitation other than conventional end-effector position control, in a hierachical order, i.e. lower priority tasks do not influence on the behavior of higher priority ones [19].

Let's consider a robotic system with n Degrees of Freedom (DoFs), being $\mathbf{q} \in \mathbb{R}^n$ the vector of configuration variables, which control input is the desired value of configuration variable time derivative $\dot{\mathbf{q}}$, and a hierarchical set n_t of tasks where $\mathbf{x}_i \in \mathbb{R}^{m_i}$, $i = 1, \cdots, n_t$ is the vector of the i-th task variables, each of those can be represented as a function of $\mathbf{q}$, i.e $\mathbf{x}_i = f_i(\mathbf{q})$ and m_i is the dimension of the i-th task. The time derivative of the i-th task variables can be written as

$$\dot{\mathbf{x}}_i = \frac{df_i(\mathbf{q})}{dt} = \mathbf{J}_i\dot{\mathbf{q}} \tag{5.0.1}$$

where $\mathbf{J}_i \in \mathbb{R}^{m_i \times n}$ is the i-th task Jacobian. The solution of eq. (5.0.1) is the minimum norm solution of the minimization problem

$$\min_{\dot{\mathbf{q}}} ||\dot{\bar{\mathbf{x}}}_i - \mathbf{J}_i\dot{\mathbf{q}}||^2 \tag{5.0.2}$$

where $\dot{\bar{\mathbf{x}}}$ is a reference velocity vector of the i-th task. This solution is given by the generalized inverse matrix $\mathbf{J}_i^{\#}$ as:

$$\dot{\mathbf{q}} = \mathbf{J}_i^{\#}\dot{\mathbf{x}}_i + (\mathbf{I} - \mathbf{J}_i^{\#}\mathbf{J}_i)\dot{\mathbf{q}}_0 \tag{5.0.3}$$

where $(\mathbf{I} - \mathbf{J}_i^{\#}\mathbf{J}_i)$ is the projection matrix with image space equal to the null space of $\mathbf{J}_i$, $\dot{\mathbf{q}}_0 \in \mathbb{R}^n$ is an arbitrary vector of null-space joint velocities. The i-th task can be than controlled by a simple a proportional controller

$$\dot{\bar{\mathbf{x}}}_i = \mathbf{K}_i(\mathbf{x}_{di} - \mathbf{x}_i) \tag{5.0.4}$$

where $\mathbf{x}_{i_d}$ is the desired value of the i-th task variables, $\mathbf{K}_i \in \mathbb{R}^{m_i \times m_i}$ is the symmetric and positive definite task gain matrix. Adding the vector $\dot{\mathbf{q}}_0$ to the solution, as it is clear in eq. (5.0.3), allows the generation of internal motions in the kinematic chain without affecting the goal of the i-th task, i.e. reaching its vector space $\mathbf{x}_i$. Hence, eq. (5.0.3) can be used again to find $\dot{\mathbf{q}}_0$ the solution of the $(i-1)$-th task having lower priority. This concept will be applied recursively on all the tasks of the task framework, so lower priority tasks do not influence on the behavior of higher priority tasks. However, the generalized inverse matrix used to solve eq. (5.0.1) creates a discontinuity when a singularity of the matrix $\mathbf{J}_i$ is encountered. Another type of discontinuity appears during the activation and the deactivation of inequality control objectives of a given task. Inequality control objective are tasks requiring to constraints of the type $\mathbf{x}_i \leq \mathbf{x}_{Mi}$ or $\mathbf{x}_i \geq \mathbf{x}_{mi}$. When $\mathbf{x}_i$ needs to stay within an interval, two separate inequality objectives can be used to represent the problem. A relevant example of inequality task is the joint limiter. An inequality task becomes active only when its control objective is going to be violated. On the other hand, even though conventional equality control tasks, such as end-effector pose control, can be always considered active, the activation and deactivation feature enables to switch among different control objectives according to the application needs. To deal with task activation and deactivation, a diagonal activation matrix $\mathbf{A}_i \in \mathbb{R}^{m_i \times m_i}$ can be considered for the i-th task. The j-th element on the diagonal of $\mathbf{A}_i$, namely $a_i(j)$, could be 1 if the j-th line of the i-th task is active, $a_i(j) = 0$ if it is inactive and $0 < a_i(j) < 1$ if in transition. Though, the original minimization problem (5.0.5) is replaced by the following

$$\min_{\dot{\mathbf{q}}} \; [||\mathbf{A}_i(\dot{\bar{\mathbf{x}}}_i - \mathbf{J}_i\dot{\mathbf{q}})||^2 + ||\mathbf{J}_i\dot{\mathbf{q}}||^2_{\mathbf{A}_i(\mathbf{I}-\mathbf{A}_i)} + ||\mathbf{V}_i^T\dot{\mathbf{q}}||^2_{\mathbf{P}_i}] \tag{5.0.5}$$

where $\mathbf{V}_i^T$ is the right orthonormal matrix of the SVD decomposition of $\mathbf{J}_i^T\mathbf{A}_i\mathbf{J}_i = \mathbf{U}_i\Sigma_i\mathbf{V}_i^T$ and $\mathbf{P}_i$ is a diagonal regularization matrix where each element $p_{(i,i)}$ is a bell-shaped function of the corresponding singular value of $\mathbf{J}_i$, or zero if the correspondent singular value do not exist and the notation $|| \cdot ||_{\mathbf{P}}$ indicates the weighted norm, i.e. $||\dot{\mathbf{q}}||^2_{\mathbf{P}_i} = \dot{\mathbf{q}}^T\mathbf{P}_i\dot{\mathbf{q}}$.

The solution of (5.0.5) can be then written as

$$\begin{aligned}\dot{\mathbf{q}} = \; & (\mathbf{J}_i^T\mathbf{A}_i\mathbf{J}_i + \mathbf{V}_i^T\mathbf{P}_i\mathbf{V}_i)^{\#}\mathbf{J}_i^T\mathbf{A}_i\mathbf{A}_i\dot{\bar{\mathbf{x}}}_i \\ & + (\mathbf{I} - (\mathbf{J}_i^T\mathbf{A}_i\mathbf{J}_i + \mathbf{V}_i^T\mathbf{P}_i\mathbf{V}_i)^{\#}\mathbf{J}_i^T\mathbf{A}_i\mathbf{A}_i\mathbf{J}_i)\dot{\mathbf{q}}_0\end{aligned} \tag{5.0.6}$$

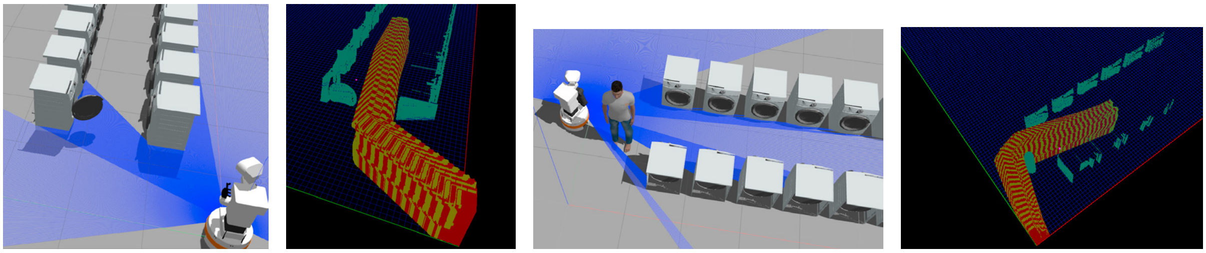
To maximize also safety and power saving capabilities by means of the proposed task priority controller, the previously described control framework is further extended by including the robot inertia matrix $\mathbf{M}$ in the computation of the generalized pseudo-inverse. Hence, the kinetic energy enters directly in the minimization problem, thus reducing the motion of the robot DoFs associated to larger inertia and leveraging more on the lighter ones, thus reducing power consumption and improving battery life. By considering this different type of discontinuities, the extension of any priority levels with the initialization $\dot{\mathbf{q}}_0 = \mathbf{0}$, $\mathbf{Q}_0 = \mathbf{I}$, for $k = 1, \cdots, n_t$, to any priority levels is expressed in the following equations

$$\begin{aligned}\mathbf{W}_k &= \mathbf{J}_k\mathbf{Q}_{k-1}(\mathbf{J}_k\mathbf{Q}_{k-1})^{\#,\mathbf{A}_k,\mathbf{Q}_{k-1},\mathbf{M}} \\ \mathbf{Q}_k &= \mathbf{Q}_{k-1}(\mathbf{I} - (\mathbf{J}_k\mathbf{Q}_{k-1})^{\#,\mathbf{A}_k,\mathbf{I},\mathbf{M}}\mathbf{J}_k\mathbf{Q}_{k-1}) \\ \mathbf{T}_k &= (\mathbf{I} - \mathbf{Q}_{k-1}(\mathbf{J}_k\mathbf{Q}_{k-1})^{\#,\mathbf{A}_k,\mathbf{I},\mathbf{M}}\mathbf{W}_k\mathbf{J}_k) \\ \dot{\mathbf{q}}_k &= \mathbf{T}_k\dot{\mathbf{q}}_{k-1} + \mathbf{Q}_{k-1}(\mathbf{J}_k\mathbf{Q}_{k-1})^{\#,\mathbf{A}_k,.,\mathbf{M}}\mathbf{W}_k\dot{\bar{\mathbf{x}}}_k\end{aligned} \tag{5.0.7}$$

$$\mathbf{x}^{\#,\mathbf{A},\mathbf{q},\mathbf{M}} \triangleq (\mathbf{M}^{-1^T}\mathbf{x}^T\mathbf{A}\mathbf{x}+\eta\mathbf{M}^{-1}(\mathbf{I}-\mathbf{Q})^T(\mathbf{I}-\mathbf{Q})+\mathbf{V}^T\mathbf{P}\mathbf{V})^{\#}\mathbf{M}^{-1^T}\mathbf{x}^T\mathbf{A}\mathbf{A} \tag{5.0.8}$$

where $\mathbf{V}$ is the right orthonormal matrix of the SVD decomposition of $\mathbf{x}^T\mathbf{A}\mathbf{x} + \eta(\mathbf{I} - \mathbf{Q})^T(\mathbf{I} - \mathbf{Q})$ and η is a suitable damping coefficient.

In our application, we considered four different tasks: the end-effector position controller task, the joint velocity minimizer task, the joint limiter task and the singularity avoidance task. To implement the singularity avoidance task, we used the concept of the manipulability index proposed in [22]. The idea is therefore to consider as control variable the smallest singular value of the robot Jacobian. Hence, by satisfying the last value, which is the smallest singular values, we are also guarantee the other singular values [21]. The joint limiter task is added in the hierarchy to ensure the safety of the robot. However, the joint velocity limiter task is added in order to reach the goal of the whole task priority while minimizing the joint speeds so the en-



(a) Environment scene in Gazebo. (b) Path planning and monitoring. (c) Human in the scene. (d) Cylindrical safety zone around the human operator.

Fig. 7: GPU-based planning and monitoring: path planning with washing machine door opened and human-aware safety zone rendered in GPU memory.

ergy spent by the robot. Since TIAGO is battery powered, the joint velocity minimizer task, together with the inertia-weighted generalized pseudoinverse, increases also increases also its battery life by reducing the overall system energy consumption.

6. Simulations and Experiments

6.1. Planning with GPU-based 3D Collision Checking

The modified navigation stack is tested in an environment with cluttered three dimensional environment scene. The test is conducted by creating an environment that has objects not detected by the laser such as open washing machine door and a suspended bar between boxes which can lead to the failure of navigation systems based on 2D obstacle rendering. For a simple comparison, two sampling based planners, namely RRT-star [7] and LBKPIECE1 [1], are chosen for the evaluation of the proposed GPU-based collision checking. Although they produced approximately identical path as shown in Fig. 7(b), planning time has improved when using control based motion planner such as LBKPIECE1. The detail of time during planning is shown with breakdown for state motion validation in Tab. 1.

Table 1: comparison of RRT-star and LBKPIECE when applying GPU collision checker

Pose and Motion validation time	LBKPIECE1	RRT-star
Total no. of Poses Inserted	674	44
Pose insertion time (ms)	3.54021	3.82593
Pose collision time (ms)	0.383763	0.463711
Poses collision	0.383763	0.463711
Total no. of motion Inserted	165	295
motion collision time (ms)	15.4182	145.363
Total Planning time in Sec.	5.188	25.48

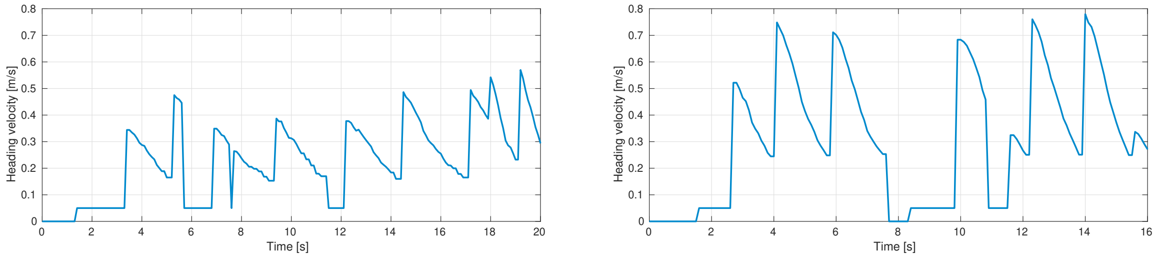
The case of successful path monitoring is also demonstrated in Fig. 7(a) and Fig. 7(b) where the robot is able to plan and monitor it's path through a corridor with an open door washing machine. The path monitoring is able to detect the particular robot pose in the plan that is in collision with the environment ahead of the actual collision through the swept volume. This effectively prune out poses of the plan that are in collision at anytime during the execution. This, in turn, results in the capability to initialize re-planning from a more suitable pose which avoids the chance that the robot get stuck in case a new obstacle in encountered along the way.

6.2. Human-Aware Safety Region Rendering

To demonstrate the proposed safety approach, the human detection and pose estimation algorithm is combined with the human-aware safety region rendering in GPU to ensure safety and collaborative behaviour. Even from partial view of the human operator, the related cylindrical safety zone can be inserted into the environment map and used during planning as shown in Fig. 7(d). This 3D-safety requirement ensures all parts of the robot(i.e. arm and mobile base) stay outside the region that collide with the safety zone during motion planning. The velocity of the robot during each execution cycle satisfies the safety requirement as shown in Fig. 8(b). The robot only reduces it's full operational velocity when it is in the human-aware mode and completely stops when the robot enters danger zone. Moreover, the validity of the remaining path segment is checked on every cycle to avoid collision and trigger re-planning ahead. The comparison of the velocity command with and without the safety constraint is shown in Fig. 8(a) and Fig. 8(b). It can be seen that the safety constraint reduced the peak velocities only in the region where human operator is in the vicinity. This allows full operational velocity when human operators are not in the scene or the distance between the operator and the robot is greater than D_{safe}.

6.3. Task priority control of the TIAGO arm

We used */arm_controller/command* interface to control the TIAGo arm in a task priority hierarchy. During our experiments, we considered 4 different tasks in the following order of priority: joint limiter task, end-effector position task, singularity avoidance task and velocity joint limiter task. After setting the different task parameters, the goal for TIAGo end-effector position control to reach the following pose (translation (x,y,z): [0.857 0.070 0.825], orientation (x,y,z): [3.033 0.079 −0.759])



(a) Velocity command with safety constraints.

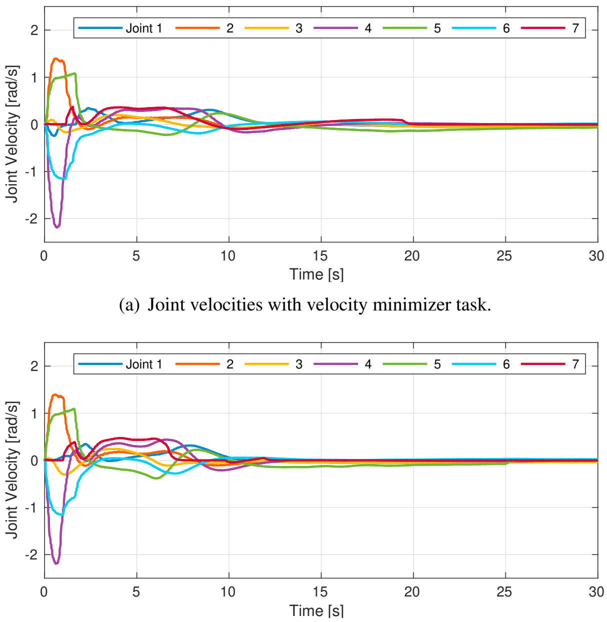
(b) Velocity command without safety constraints.

Fig. 8: Linear velocity Plot with and without Safety Requirement.

is passed. In 6.3.1, we show the effects of the velocity minimizer task during the motion of TIAGo arm between from its initial position to the end-effector goal position. However, in 6.3.2, we present the influence of using the inertia-weighted generalized pseudoinverse on the kinetic energy of the TIAGo arm.

6.3.1. *Effects of the Velocity Minimizer Tasks*

Figure 9 compares the joint velocities of TIAGo arm for two different types of task priority control where the velocity minimizer task is present and absent. Below, we can see that adding the joint velocity minimizer task reduces the joint velocities: the joint velocities of different joints is lower when the velocity minimizer task is present 9(a) compared to when it is absent 9(b). This conclusion is clearly seen in Table 2 and 3.



(a) Joint velocities with velocity minimizer task.

(b) Joint velocity without velocity minimizer task.

Fig. 9: Effects of the velocity minimizer task during the TIAGo arm motion.

Table 2 shows how the root mean square (RMS) velocities of all the joints reduced when joint velocity minimizer task is added compared to the joint velocities without the minimizer task. Table 3 also indicate that the time required to reach a steady state velocities lower than 0.02 rad/s. A reduced joint velocities implies less energy consumption, which improves the battery life of the TIAGo robot.

Figure 11 shows different steps of the robot motion starting from its initial position and reaching the end-effector goal.

6.3.2. *Effects of the Inertia Matrix in Task Priority Control*

Figure 10 compares the kinetic energy of TIAGo's arm during the motion task with and without inertia-weighted generalized pseudoinverse. The task priority hierarchy considered the 4 different tasks that are already introduced and the robot gripper should reach the latter defined goal. It is clear that the robot kinetic energy is lower when the inertia-weighted pseudoinverse is used (see Figure 10). Though, this solution allows to save energy while guaranteeing.

7. Conclusion and Future work

This work has presented the results on appliance test automation with the introduction of mobile manipulators in the test laboratory environment. The overall system is conceived to operate in a safe way in the vicinity of human operator while also optimizing energy usage of the arm.

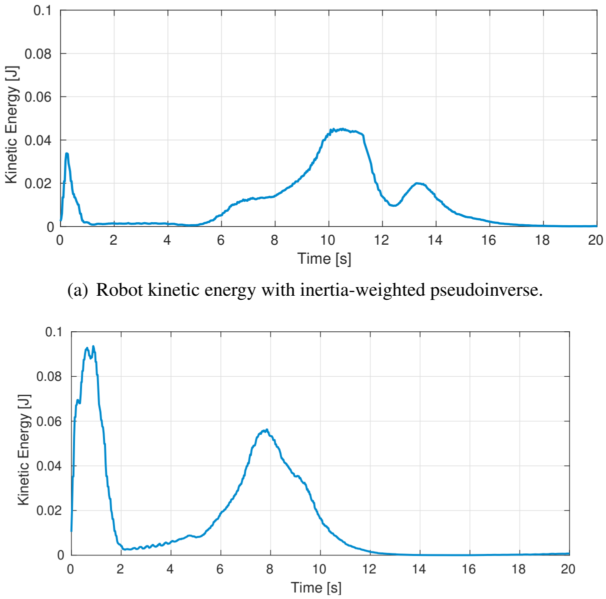
The experimental and simulation tests are conducted for all the subsystems considering a scenario similar to the actual test laboratory. The result of the experiment showed significant success in navigating through narrow corridors while maintaining correct detection of appliances and estimating their pose both from close and far range.

Joint	1	2	3	4	5	6	7
With	0.080	0.169	0.05	0.240	0.174	0.160	0.100
Without	0.082	0.172	0.065	0.243	0.180	0.165	0.114

Table 2: Effect on the velocity minimizer on the RMS value of joint velocities.

Joint	1	2	3	4	5	6	7
With	10.50	12.71	10.90	12.62	25.30	13.83	12.09
Without	12.17	14.34	13.39	15.07	34.45	18.62	19.73

Table 3: Time required for each joint to reach a velocity less than 0.02 rad/s



(a) Robot kinetic energy with inertia-weighted pseudoinverse.

(b) Robot kinetic energy without inertia-weighted pseudoinverse.

Fig. 10: Kinetic energy of the TIAGo arm during the motion task with and without inertia-weighted generalized pseudoinverse.

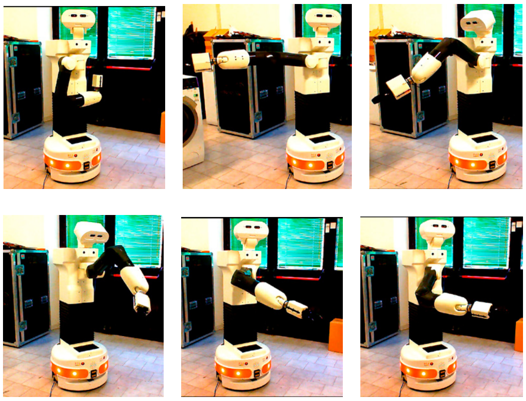


Fig. 11: Different steps of the motion of TIAGo arm.

Acknowledgments

This work was partially supported by the European Commission's Horizon 2020 Framework Programme with the project REMODEL - Robotic technologies for the manipulation of complex deformable linear objects - under grant agreement No 870133.

References

[1] Bohlin, R., Kavraki, L.E., 2000. Path planning using lazy prm, in: Proceedings 2000 ICRA. Millennium Conference. IEEE Int. Conf. on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065), pp. 521–528.

[2] Choi, B.J., Jin, S.M., Shin, et al., 2008. Development of flexible laboratory automation platform using mobile agents in the clinical laboratory, in: Proc. IEEE Int. Conf. on Automation Science and Engineering, pp. 918–923.

[3] Escobar, L.A., Meeker, W.Q., et al., 2006. A review of accelerated test models. Statistical science 21, 552–577.

[4] Hamner, B., Koterba, S., Shi, J., Simmons, R., Singh, S., 2010. An autonomous mobile manipulator for assembly tasks. Autonomous Robots 28, 131.

[5] Hermann, A., Bauer, J., Klemm, S., Dillmann, R., 2014. Mobile manipulation planning optimized for gpgpu voxel-collision detection in high resolution live 3d-maps, in: Int. Symp. on Robotics, pp. 1–8.

[6] Hermann, A., Klemm, S., Xue, Z., Roennau, A., Dillmann, R., 2013. Gpu-based real-time collision detection for motion execution in mobile manipulation planning, in: Proc. IEEE Int. Conf. on Advanced Robotics, pp. 1–7.

[7] Karaman, S., Frazzoli, E., 2011. Sampling-based algorithms for optimal motion planning. The Int. Journal of Robotics Research 30, 846–894.

[8] Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L., 2014. Microsoft coco: Common objects in context, in: European conference on computer vision, Springer. pp. 740–755.

[9] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S.E., Fu, C.Y., Berg, A.C., 2016. Ssd: Single shot multibox detector, in: ECCV.

[10] Marvel, J.A., 2013. Performance metrics of speed and separation monitoring in shared workspaces. IEEE Tran. on Automation Science and Engineering 10, 405–414.

[11] Nooruddin, F.S., Turk, G., 2003. Simplification and repair of polygonal models using volumetric techniques. IEEE Tran. on Visualization and Computer Graphics 9, 191–205.

[12] O'Mahony, N., Campbell, S., Carvalho, A., Harapanahalli, S., Hernandez, G.V., Krpalkova, L., Riordan, D., Walsh, J., 2019. Deep learning vs. traditional computer vision, in: Science and Information Conference, Springer. pp. 128–144.

[13] Pages, J., Marchionni, L., Ferro, F., 2016. Tiago: the modular robot that adapts to different research needs, in: Int. workshop on robot modularity, IROS.

[14] Pan, J., Chitta, S., Manocha, D., 2012. Fcl: A general purpose library for collision and proximity queries, in: Proc. IEEE Int. Conf. on Robotics and Automation, pp. 3859–3866.

[15] Pan, J., Lauterbach, C., Manocha, D., 2010. g-planner: Real-time motion planning and global navigation using gpus, in: Twenty-Fourth AAAI Conference on Artificial Intelligence.

[16] Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Wheeler, R., Ng, A.Y., 2009. Ros: an open-source robot operating system, in: ICRA workshop on open source software, Kobe, Japan. p. 5.

[17] Sandler, M., Howard, A.G., Zhu, M., Zhmoginov, A., Chen, L.C., 2018. Mobilenetv2: Inverted residuals and linear bottlenecks. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition , 4510–4520.

[18] Shea, R.N., 2013. Robot safety standard update, in: RIA meeting, pp. 20–23.

[19] Simetti, E., Casalino, G., 2016. A novel practical technique to integrate inequality control objectives and task transitions in priority based control. Journal of Intelligent & Robotic Systems 84, 877–902.

[20] Sucan, I.A., Moll, M., Kavraki, L.E., 2012. The open motion planning library. IEEE Robotics & Automation Magazine 19, 72–82.

[21] Sverdrup-Thygeson, J., Moe, S., Pettersen, K.Y., Gravdahl, J.T., 2017. Kinematic singularity avoidance for robot manipulators using set-based manipulability tasks, in: 2017 IEEE Conference on Control Technology and Applications (CCTA), pp. 142–149.

[22] Yoshikawa, T., 1985. Manipulability and redundancy control of robotic mechanisms, in: Proceedings. 1985 IEEE Int. Conf. on Robotics and Automation, pp. 1004–1009.