The Autonomous Picking & Palletizing (APPLE) Robot: A Research Platform for Intralogistics Applications

Robert Krug*, Todor Stoyanov*, Vinicio Tincani‡, Henrik Andreasson*, Rafael Mosberger*, Gualtiero Fantoni‡, Achim J. Lilienthal* and Antonio Bicchi‡

***Abstract*— Todo ...**

I. Introduction

The increasing need for fast and flexible commissioning (*i.e.*, order picking and collection of unstructured goods from storage compartments in warehouses) in logistic scenarios has created substantial interest for autonomous robotic solutions. One of the main arguments for automating this task is that the dull and strenuous nature of commissioning could cause mental and physical illness in human workers. As a result, the determination within the logistics sector to invest in this area is high and substantial efforts are made for the humanization of workstations [1].

There exist partial solutions for the automated commissioning problem in controlled environments. The system described in [2] coordinates a fleet of Autonomous Ground Vehicles (AGVs) which transport shelves filled with goods to a human worker who picks the corresponding objects to complete the order. Key obstacles for a fully automatized solution applicable in general warehouse settings are the safe autonomous vehicle navigation in industrial environments co-populated by humans, as well as the autonomous grasping/manipulation of unstructured goods at reasonable cycle times.

In this work, we present the Autonomous Picking & Palletizing (APPLE) research platform (see Fig. 1) which we developed to address the following important sub-task chain which occurs during commissioning in prototypical warehouses: autonomous picking of goods from a storage location, subsequent placement on a standard EUR half-pallet and transport of the filled pallet to a target location. Furthermore, this process has to be carried out in a manner which is safe for humans operating in the same environment. The platform's mobile base consists of a non-holonomic forklift AGV which is able to detect and pick up pallets in designated loading zones @Henrik: is that a reasonable one-sentence description...?. A KUKA LBR iiwa leight-weight arm which is fitted with an under-actuated gripper with conveyor belts on the inside of each finger is used for robust grasping and object manipulation. While this setup is not intended as a close-to-market solution for palletizing, the force/torque sensing capabilities as well as the compliant and human-safe control possibilities allow us to investigate various grasping/manipulation strategies.

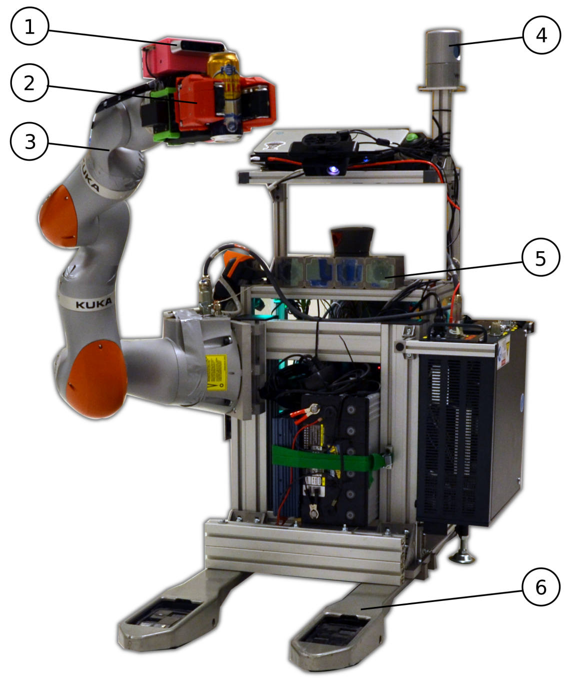


Fig. 1. *The APPLE platform:* A KUKA LBR iiwa arm (3) is mounted on a retrofitted Linde CitiTruck AGV (6). For localization, a Velodyne Lidar (4) is used, human worker detection is carried out with the RefleX camera system (5). The depicted grasping device (2) is a further developed and smaller version of the Velvet Fingers gripper described in [3]. Each of the grippers two fingers has a planar manipulator structure with two rotary joints and active surfaces which are implemented by conveyor belts on the inside of the two phalanges. The mechanical structure of each finger is underactuated and comprises one actuated Degree of Freedom (DoF) for opening and closing and two DoF for the belt movements. If, during grasping, the proximal phalanges are blocked by an object, the grippers distal phalanges continue to wrap around and envelope it in a firm grasp. Object and pallet detection is done with an ASUS Xtion Pro camera (1) which is mounted on the gripper's palm.

In this paper, we outline our solution to the safe navigation of the APPLE platform in an industrial scenario co-populated by human workers wearing reflective vests. Furthermore, we introduce a novel grasp planning- and representation scheme which is utilized for reactive, on-the-fly manipulator motion generation. This allows to exploit manipulator redundancies and offers several advantages to the commonly used sense-plan-act architectures. As a final contribution, we discuss our compliant grasp execution strategy which uses the active surfaces on the employed gripper to increase grasp robustness.

The remainder of this article is organized as follows:

* AASS Research Center; Örebro University; Studentgatan 1, 70182 Örebro, Sweden.
‡ Interdepart. Research Center "E. Piaggio"; University of Pisa, Via Diotisalvi 2, 56100 Pisa, Italy.

below we outline the AGV navigation and motion planning scheme before presenting our solution for people detection in Section II-A. In Section III our approach to robust, online grasp motion generation and grasp execution is discussed. We show early results and an evaluation of the APPLE system by means on a simplified commissioning task in Section IV before drawing conclusions in Section V.

II. Autonomous Forklift

Introduce the CitiTruck + RefleX camera system

A. Related Work

B. Navigation

This module ensures that the AGV is capable to move autonomously and safely through the workspace environment. In order to achieve this task, we use components of a navigation system previously developed in the context of our KKS-funded Safe Autonomous Vehicles (SAUNA) project. We construct a 3D map of the static parts of the environment (using [4]) and use it to localize the vehicle in the presence of dynamic entities (using [5]). For motion planning and control of the non-holonomic AGV, we will our lattice planner [6] and a model-predictive tracking controller. The complete navigation system has been implemented, extensively tested and successfully integrated on the APPLE demonstrator, a detailed description can be found in [7].

C. People Detection

As the envisioned mobile manipulation system will operate in environments shared with human workers, people detection and human safety are important issues. In APPLE we address the problem by using the RefleX system we recently developed [8]. RefleX is a camera-based on-board safety system for industrial vehicles and machinery for detection of human workers wearing reflective vests worn as per safety regulations. The system was designed with industrial safety standards in mind and is currently being tested as an industrial prototype.

III. Grasping & Manipulation System

In this section we present an integrative approach for grasp representation and motion generation. The main idea is to provide a functional representation of grasps as intervals in task space, which allows redundancy in the gripper pose prior to executing the grasp. Subsequently, we leverage the obtained redundancy to generate reactive manipulator motions by formulating a representation of the corresponding tasks in the prioritized motion control framework proposed by Kanoun et al. [9].

A. Related Work

In current autonomous grasping systems [10], [11], [12], grasp planning and manipulator motion planning are usually seen as independent sub-problems [13]. A database storing object models together with sets of pre-computed grasps is used to find suitable gripper poses/configurations [14], [15], [12]. In the online stage, sampling based planners [16] attempt to generate valid trajectories for the pre-planned grasps, which are executed in a feasible-first manner [10]. During the execution phase, such approaches necessitate many futile motion planning attempts, which often incurs significant time delays since sampling based planners suffer from the curse of dimensionality. Also, while being able to solve complicated planning problems if given enough time, these planners do not scale well to geometrically simple scenarios and they are ill suited to incorporate contact events with the environment which is a prerequisite for any grasping/manipulation application.

Using control to exploit manipulator redundancy has long been in the focus of research [17], [18]. In this line of thought, opposed to define a set of discrete grasp poses, Gienger et al. [19], [20] learn a manifold of feasible grasps for an object and use an attractor-based control scheme to reach the corresponding volume in the task space. Another approach to address motion planning and motion control simultaneously, is to formulate a hierarchical stack of equality tasks which are represented as task functions [21]. Lower-ranked tasks are then solved in the null-space of tasks with higher priority [17], [18]. A major step forward in this direction was achieved by Kanoun et al. [9], who extended the approach to include inequality tasks which significantly increases the expressiveness of the method. Recent advancements include the extension to dynamic control [22] and an efficient solution algorithm for the underlying optimization problem [23].

B. The Grasp Interval

Traditional data-driven grasp synthesis [24] relies on precise knowledge of the target object geometry and frictional properties. Since this information is often not available in practice, grasps planned in simulation and under these assumptions might fail. This holds especially true, when considering an underactuated grasping device as we do in the presented work. In this case, the joint configuration depends on the interaction with the environment and is difficult or impossible to determine at planning time. Instead, we represent grasps as predefined intervals associated with primitive object geometries as exemplary shown in Fig. 2 for cylindrical objects. These intervals bound the grasping device's position and orientation but do not fully constrain its pose. For simplicity we limit ourselves to the illustrated grasp interval defined for cylindrical shapes, corresponding intervals can be defined for other shape categories such as spheres and parallelepipeds. Opposed to the similarly defined task intervals in [19], [20], which are learned in simulation, we explicitly incorporate prior knowledge about robust grasp poses in our representation. In a concept originating from observations of human grasping behavior, it has been shown that the grasping device should be roughly aligned with the target object's principal components to achieve robust grasps [25]. This property is achieved by the cone constraints for the exemplary case depicted in Fig. 2.

Currently, the parameters of the grasp intervals such as the distance range between gripper and object have to be

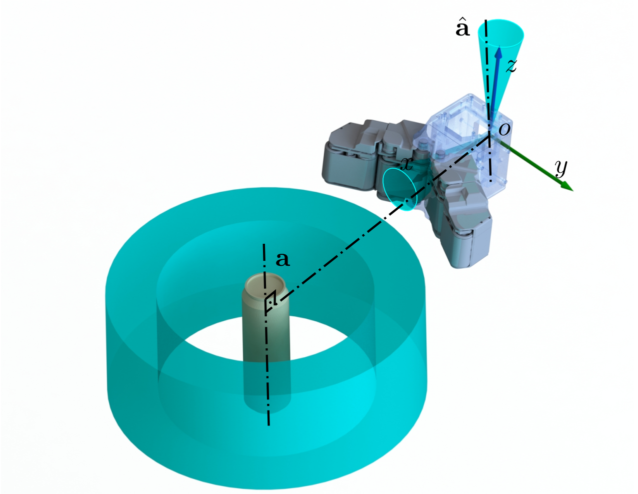


Fig. 2. *Grasp interval:* The shaded cyan regions illustrate the grasp interval for a cylindrical object. For a successful grasp, the palm frame origin $\boldsymbol{o}$ needs to lie inside the depicted cylindrical shell which is aligned with object axis $\boldsymbol{a}$. The cylinders height is limited by two planes (not depicted) which are normal to $\boldsymbol{a}$. Additionally, the gripper's vertical axis (z) is constrained to lie in a cone whose axis $\hat{\boldsymbol{a}}$ is parallel to the object axis $\boldsymbol{a}$, and the gripper's approach axis (x) has to lie inside a cone centered on the normal connecting axis $\boldsymbol{a}$ and point $\boldsymbol{o}$.

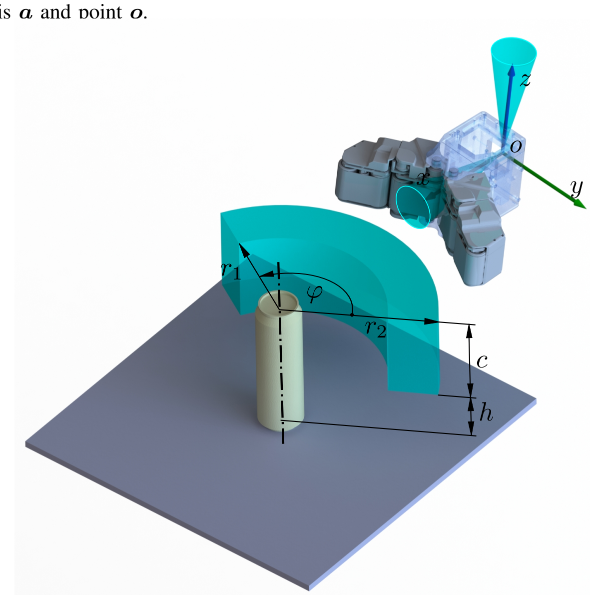


Fig. 3. *Truncated grasp interval:* todo...

evaluated experimentally for each primitive shape category. To ease this non-trivial requirement, in the presented work we rely on a gripper which offers a low pre-grasp pose sensitivity combined with a compliant and robust grasp execution routine as discussed in Section III-E.

C. Object Detection and Grasp Planning

Typical commissioning scenarios in a warehouse environment would generally involve as a first step the identification and localization of an object targeted for picking. There are however many application scenario specific factors to consider when designing an object detection system for this task. For example: a single pallet may hold a homogeneous or a heterogeneous set of objects; objects may be stored in boxes and equipped with barcodes; objects may be stored on shelves, pallets, or simply piled up. For the APPLE project we assumed a simple scenario where objects of the same type are stored on a single pallet. We have equipped the Velvet Fingers 2 gripper with a structured light depth sensor: the Structure IO[1] device. We use the 3D point clouds generated by the sensor as an input for a simple object detection module which identifies clusters of points matching our expected object models. The object detection module was implemented using the point cloud library (PCL)[2] following the simple algorithm described in Alg. 1.

Algorithm 1: Object detection algorithm

input : Pointcloud P, kinematic model K, joint positions $\boldsymbol{j}$
output: Target cluster T
Transform P into world coordinates using $K, \boldsymbol{j}$;
while *target plane not found* **do**
  Extract plane $\boldsymbol{n}, d$ from P using RANSAC;
  if $\boldsymbol{nz} < \cos\alpha, abs(d - h_{pallet}) < e_{pallet}$ **then**
    target plane found;
  end
  else
    Remove inliers and iterate. If points in P less than 20% of original points, report failure;
  end
end
Extract oriented bounding box (OBB) of plane inliers;
Extract points inside OBB and height in $(d, d + d_{eps})$;
Euclidean cluster points in P_n;
for *each cluster* E_i **do**
  Fit a ML Gaussian to E_i as μ_i, Σ_i;
  Eigen decomposition $\Sigma_i = Q \Lambda Q^{-1}$;
  Let $\boldsymbol{q}_1$ be the eigenvector associated to the largest eigenvalue λ_1;
  if $\boldsymbol{q}_1 \boldsymbol{z} < \cos\alpha$ *and* $\lambda_2 < r_{max}$ **then**
    report target cluster at μ_i, Σ_i;
  end
end

The basic idea of Alg. 1 is to first detect the location of the pallet plane, then to extract the points belonging to objects on the pallet, cluster them and then analyze the obtained clusters. The parameters in this algorithm are: h_{pallet} — the expected height of the pallet above floor level; e_{pallet} — a tolerance of the pallet height; α — a tolerance on the angle that the pallet plane normal makes with the vertical axis $\boldsymbol{z}$; r_{max} — the maximum radius of a graspable cylinder. Each of the detected clusters is analyzed using PCA, checking that the dominant point distribution is along the $\boldsymbol{z}$ axis and that the $x - y$ point spread is within the graspable objects limit. Identified target clusters are then passed on to the grasp

[1] `http://structure.io/`
[2] `http://pointclouds.org/`

planning module. @Todor: should the cylinder shell fitting be cited here ...? [26]

D. Manipulator Motion Generation

We lean on the notation in [23] and define the manipulator joint configuration by the vector $\boldsymbol{q}$ and the control inputs as corresponding joint velocities $\dot{\boldsymbol{q}}$. A task function is any derivable function of $\boldsymbol{e}$. To give an example, a task with the purpose of bringing an end-effector point $\boldsymbol{p}(\boldsymbol{q})$ onto a plane described by unit normal $\boldsymbol{n}$ and offset b can be transcribed by the task function $\boldsymbol{e} = \boldsymbol{n}^T\boldsymbol{p}(\boldsymbol{q}) - d$, which formulates the projection residual between the plane and $\boldsymbol{p}(\boldsymbol{q})$ connect example to the previously described grasp interval. The task evolution is given by $\boldsymbol{J}\dot{\boldsymbol{q}} = \dot{\boldsymbol{e}}$ with task jacobian $\boldsymbol{J} = \frac{\partial \boldsymbol{e}}{\partial \boldsymbol{q}}$.

Goal is to compute joint velocities such that the task evolution follows a desired reference profile $\dot{\boldsymbol{e}}^*$ (often chosen as exponential decay $\boldsymbol{e}^* = -\lambda \boldsymbol{e}$, with $\lambda \in \mathbb{R}_+$). For a single equality task, this necessitates to solve the following least squares Quadratic Program (QP)

$$\dot{q}^* \in \arg\min_{\dot{\boldsymbol{q}}} ||\boldsymbol{J}\dot{\boldsymbol{q}} - \dot{\boldsymbol{e}}^*||. \tag{1}$$

In order to allow for inequality tasks, we henceforth use a general task formulation with upper bounds

$$\boldsymbol{J}\dot{\boldsymbol{q}} \leq \boldsymbol{e}^* \tag{2}$$

as in [23]. This allows to transcribe lower bounds $\boldsymbol{J}\dot{\boldsymbol{q}} \geq \boldsymbol{e}^*$, double bounds $\underline{\dot{\boldsymbol{e}}}^* \leq \boldsymbol{J}\dot{q} \leq \bar{\dot{\boldsymbol{e}}}^*$ and equalities $\boldsymbol{J}\dot{\boldsymbol{q}} = \boldsymbol{e}^*$ by re-formulating the task respectively as $-\boldsymbol{J}\dot{\boldsymbol{q}} \leq -\boldsymbol{e}^*$, $\begin{bmatrix} -\boldsymbol{J} \\ \boldsymbol{J} \end{bmatrix} \dot{\boldsymbol{q}} \leq \begin{bmatrix} -\underline{\dot{\boldsymbol{e}}}^* \\ \bar{\dot{\boldsymbol{e}}}^* \end{bmatrix}$ and $\begin{bmatrix} -\boldsymbol{J} \\ \boldsymbol{J} \end{bmatrix} \dot{\boldsymbol{q}} \leq \begin{bmatrix} -\dot{\boldsymbol{e}}^* \\ \dot{\boldsymbol{e}}^* \end{bmatrix}$.

If the constraint $\boldsymbol{J}\dot{\boldsymbol{q}} \leq \boldsymbol{e}^*$ is infeasible, a least squares solution for $\boldsymbol{q}^*$ as in (1) can be found by introducing the slack variable $\boldsymbol{w}$ in the decision variables

$$\begin{aligned} \min_{\dot{\boldsymbol{q}},\, \boldsymbol{w}} \quad & ||\boldsymbol{w}|| \\ \text{subject to} \quad & \boldsymbol{J}\dot{\boldsymbol{q}} \geq \boldsymbol{e}^* + \boldsymbol{w}. \end{aligned} \tag{3}$$

To form a stack of hierarchical tasks with $p = 1, \ldots, P$ priority levels, we stack all task jacobians in (2) with the same assigned priority in a matrix $\boldsymbol{A}_p$, and all corresponding reference velocities in a vector $\boldsymbol{b}_p$ to form a constraint of the form $\boldsymbol{A}_p\dot{\boldsymbol{q}} \leq \boldsymbol{b}_p$ for each hierarchy level. The aim is to sequentially satisfy a constraint at best in the least square sense while solving for the subsequent constraint of lower priority in the null-space of the previous constraint, such that the previous solution is left unchanged. Therefore, the following QP, where the previous slack variable solutions $\boldsymbol{w}_i^*$ are frozen between iterations, needs to be solved for $p = 1, \ldots, P$

$$\begin{aligned} \min_{\dot{\boldsymbol{q}},\, \boldsymbol{w}_p} \quad & ||\boldsymbol{w}_p|| \\ \text{subject to} \quad & \boldsymbol{A}_i\dot{\boldsymbol{q}} \leq \boldsymbol{b}_i + \boldsymbol{w}_i^*, \quad i = 1, \ldots, p-1 \\ & \boldsymbol{A}_p\dot{\boldsymbol{q}} \leq \boldsymbol{b}_p + \boldsymbol{w}_p. \end{aligned} \tag{4}$$

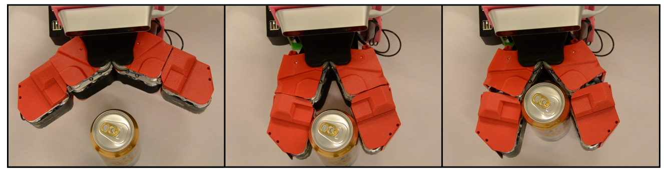


Fig. 4. *Pull-in grasping strategy:* Depicted is a sequence of intermediate grasp states where the belts of the gripper are used to pull the object towards its palm which results in a transition from a fingertip to an enveloping grasp.

The final joint velocity vector $\boldsymbol{q}^*$ is obtain from the P^{th} solution of (4).

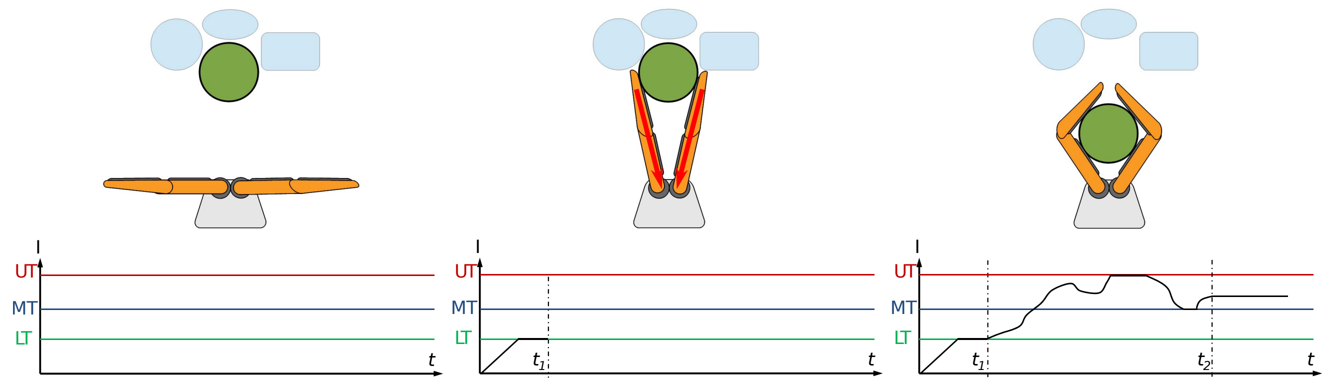
For reactive on-the-fly motion generation we formulate a stack of hierarchical tasks and use the recently developed method by Kanoun et al. [9], which allows to account for inequality tasks and solves a sequence of convex optimization problems at each time step to obtain appropriate joint velocity commands.

Obstacle avoidance is also achieved on a control-level, by formulating tasks which maintain minimum distances between simple geometric primitives such as spheres, planes, points and capsules. We argue that for the considered application strict collision avoidance is neither necessary nor desired, since picking and manipulation inherently necessitates contact events between the robot and the environment. Also, in real-world applications where knowledge about the environment is available only in form of noisy sensor data, it might not be possible to avoid contacting the environment without being overly conservative. This makes the KUKA LBR iiwa with its compliant low-level control schemes and contact detection abilities an ideal platform for the tackled purpose and motion generation scheme. The relatively simple picking task in APPLE provides an ideal testbed in a real-world scenario.

E. Robust Grasp Execution

For this component, we leverage the capabilities of the Velvet Gripper, namely underactuation and conveyor belts on the finger pads in order to achieve robust grasping behavior. Especially in cluttered scenes, a "pull-in" strategy has been shown to be especially effective to achieve stable grasps while starting from a relatively wide range of initial gripper poses with respect to the target object [12]. Here, the features of the grasping device are exploited to embrace the object in a firm envelope grasp by simultaneously squeezing it in a compliant fashion while actuating the belts inwards.

(the following is copy/pasted from the Gripper control workshop paper) [27] Each of the grippers two fingers has a planar manipulator structure with two joints and active surfaces which are implemented by coupled conveyor belts on the inside of the two phalanges. The mechanical structure is underactuated and comprises only one actuated Degree of Freedom (DoF) for opening and closing and two DoF for the belt movements. If, during grasping, the proximal phalanges are blocked by an object, the grippers distal phalanges continue to wrap around and envelope it in a firm grasp. The experiments reported in [12] showed, that in cluttered scenes fingertip grasps are more likely to be feasible



(a) (b) (c)

Fig. 5. *Grasp Execution Control:* As the open griper closes in on the object (Left), the current through the opening motor is monitored. When contact is made (Middle), the actuated belts are switched on to pull in the object. The controller then strives to maintain the current in between two target thresholds by opening or closing the gripper during in-hand manipulation (Right).

than robust enveloping grasps, because the latter necessitate large opening angles resulting in bulky gripper silhouettes for which no collision free approach trajectories can be found. Therefore, we employ the pull-in strategy which is illustrated in Fig. **??**. Here, the underactuated nature and the conveyor belts on the grasping device are exploited to embrace the object in a firm envelope grasp by simultaneously squeezing it while actuating the belts inwards. This is achieved by compliant low-level position controllers which saturate on experimentally evaluated current thresholds. We use a simple grasping routine which is triggered after an initial fingertip grasp is achieved (see Fig. **??**). This routine consists of issuing commands to fully close the gripper while moving the belts a pre-defined offset towards the palm. Three thresholds on the current absorption of the opening motor are used: a low threshold (LT) signifies contact between the gripper and the object and a mid threshold (MT) indicates a large enough contact force to stop the closing movement. Finally, an upper threshold (UT) prevents damage to the grasping device. Once the pull-in sequence is completed, the controllers maintain the final torques to ensure a stable grasp.

IV. Evaluation

commands while accounting for the robot dynamics [22])

The aim is to reduce the dependence on classical, sampling based motion planning and to move towards reactive feedback control to generate and execute complex motion behaviors of a robot. Here, only high-level behavioral goals (e. g., go to this region or stay above obstacle plane) are specified in form of task functions [21]. An intelligent control algorithm, which is based on embedded optimization of these task functions, then handles the details and synthesizes appropriate motions automatically in an online fashion. Opposed to classical sense-plan-act architectures, in this paradigm only task-relevant Degrees of Freedom (DoF) need to be constrained, which allows to exploit kinematic redundancies, e. g., for a manipulator to avoid unexpected obstacles. Regarding grasp planning, we follow the general

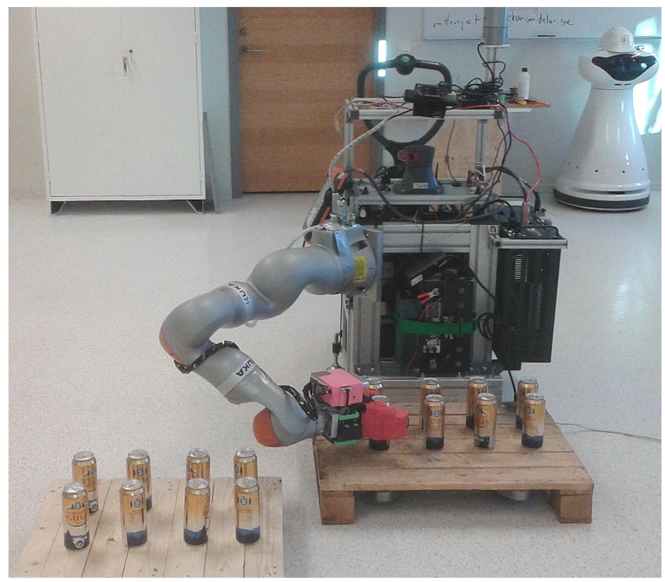


Fig. 6. *Demo setup:* The robot loads beer onto a previously picked up pallet and subsequently transports it to a predefined target location.

tenet and will extract redundant representations in form of constrained pose intervals instead of discrete poses

[28] task function description [29](Used off-the-shelf solver) [30](ROS)

V. Discussion and Outlook

[31][32](optimal control for motion generation)

References

[1] W. Echelmeyer, A. Kirchheim, and E. Wellbrock, "Robotics-logistics: Challenges for automation of logistic processes," in *Proc. of the IEEE International Conference on Automation and Logistics*, 2008, pp. 2099–2103.

[2] P. R. Wurman, R. D'Andrea, and M. Mountz, "Coordinating hundreds of cooperative, autonomous vehicles in warehouses," *AI magazine*, vol. 29, no. 1, pp. 9–20, 2008.

[3] V. Tincani, M. G. Catalano, E. Farnioli, M. Garabini, G. Grioli, G. Fantoni, and A. Bicchi, "Velvet fingers: A dexterous gripper with active surfaces," in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012, pp. 1257–1263.

[4] T. Stoyanov, J. P. Saarinen, H. Andreasson, and A. J. Lilienthal, "Normal distributions transform occupancy map fusion: Simultaneous mapping and tracking in large scale dynamic environments," in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2013, pp. 4702–4708.

[5] R. Valencia, J. Saarinen, H. Andreasson, J. Vallve, J. Andrade-Cetto, and A. Lilienthal, "Localization in highly dynamic environments using dual-timescale ndt-mcl," in *Proc. of the IEEE International Conference on Robotics and Automation*, May 2014, pp. 3956–3962.

[6] M. Cirillo, T. Uras, S. Koenig, H. Andreasson, and F. Pecora, "Integrated motion planning and coordination for industrial vehicles," in *Proc. of the International Conference on Automated Planning and Scheduling*, 2014.

[7] H. Andreasson, J. Saarinen, M. Cirillo, T. Stoyanov, and A. J. Lilienthal, "Fast, continuous state path smoothing to improve navigation accuracy," in *Proc. of the IEEE International Conference on Robotics and Automation*, 2015, to appear.

[8] R. Mosberger, H. Andreasson, and A. J. Lilienthal, "A customized vision system for tracking humans wearing reflective safety clothing from industrial vehicles and machinery," *Sensors*, vol. 14, no. 10, pp. 17 952–17 980, 2014.

[9] O. Kanoun, F. Lamiraux, and P.-B. Wieber, "Kinematic control of redundant manipulators: Generalizing the task-priority framework to inequality task," *IEEE Transactions on Robotics*, vol. 27, no. 4, pp. 785–792, 2011.

[10] D. Berenson, R. Diankov, K. Nishiwaki, S. Kagami, and J. Kuffner, "Grasp planning in complex scenes," in *Proc. IEEE/RAS International Conference on Humanoid Robots*, 2007, pp. 42–48.

[11] S. Srinivasa, D. Ferguson, C. Helfrich, D. Berenson, A. Collet, R. Diankov, G. Gallagher, G. Hollinger, J. Kuffner, and M. VandeWeghe, "Herb: A home exploring robotic butler," *Autonomous Robots*, vol. 28, no. 1, pp. 5–20, 2010.

[12] R. Krug, T. Stoyanov, M. Bonilla, V. Tincani, N. Vaskevicius, G. Fantoni, A. Birk, A. J. Lilienthal, and A. Bicchi, "Velvet fingers: Grasp planning and execution for an underactuated gripper with active surfaces," in *Proc. of the IEEE International Conference on Robotics and Automation*, 2014, pp. 3669–3675.

[13] R. Diankov, "Automated construction of robotic manipulation programs," Ph.D. dissertation, Carnegie Mellon University, Robotics Institute, 2010.

[14] A. T. Miller and P. K. Allen, "Graspit! a versatile simulator for robotic grasping," *IEEE Robotics and Automation Magazine*, vol. 11, no. 4, pp. 110–122, 2004.

[15] C. Goldfeder and P. K. Allen, "Data-driven grasping," *Autonomous Robots*, vol. 31, no. 1, pp. 1–20, 2011.

[16] S. M. LaValle, *Planning Algorithms*. Cambridge University Press, 2006.

[17] B. Siciliano and J.-J. Slotine, "A general framework for managing multiple tasks in highly redundant robotic systems," in *Proc. of the International Conference on Advanced Robotics*. IEEE, 1991, pp. 1211–1216.

[18] L. Sentis, J. Park, and O. Khatib, "Compliant control of multicontact and center-of-mass behaviors in humanoid robots," *IEEE Transactions on Robotics*, vol. 26, no. 3, pp. 483–501, 2010.

[19] M. Gienger, M. Toussaint, and C. Goerick, "Task maps in humanoid robot manipulation," in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2008, pp. 2758–2764.

[20] M. Gienger, M. Toussaint, N. Jetchev, A. Bendig, and C. Goerick, "Optimization of fluent approach and grasp motions," in *Proc. IEEE/RAS International Conference on Humanoid Robots*, 2008, pp. 111–117.

[21] C. Samson, B. Espiau, and M. L. Borgne, *Robot control: the task function approach*. Oxford University Press, 1991.

[22] L. Saab, O. E. Ramos, F. Keith, N. Mansard, P. Soueres, and J. Fourquet, "Dynamic whole-body motion generation under rigid contacts and other unilateral constraints," *IEEE Transactions on Robotics*, vol. 29, no. 2, pp. 346–362, 2013.

[23] A. Escande, N. Mansard, and P.-B. Wieber, "Hierarchical quadratic programming: Fast online humanoid-robot motion generation," *International Journal of Robotics Research*, 2014.

[24] J. Bohg, A. Morales, T. Asfour, and D. Kragic, "Data-driven grasp synthesisa survey," *IEEE Transactions on Robotics*, vol. 30, no. 2, pp. 289–309, 2014.

[25] R. Balasubramanian, L. Xu, P. Brook, J. Smith, and Y. Matsuoka, "Physical human interactive guidance: Identifying grasping principles from human-planned grasps," *IEEE Transactions on Robotics*, vol. 28, no. 4, pp. 899–910, 2012.

[26] A. t. Pas and R. Platt, "Localizing grasp affordances in 3-d points clouds using taubin quadric fitting," *arXiv preprint arXiv:1311.3192*, 2013.

[27] R. Krug, T. Stoyanov, M. Bonilla, V. Tincani, N. Vaskevicius, G. Fantoni, A. Birk, A. J. Lilienthal, and A. Bicchi, "Improving grasp robustness via in-hand manipulation with active surfaces," in *IEEE International Conference on Robotics and Automation - Workshop on Autonomous Grasping and Manipulation: An Open Challenge*, 2014.

[28] O. Kanoun, "Contribution à la planification de mouvement pour robots humanoïdes," Ph.D. dissertation, l'Université Toulouse III - Paul Sabatier, 2009.

[29] I. Gurobi Optimization, "Gurobi optimizer reference manual," 2015. [Online]. Available: http://www.gurobi.com

[30] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "ROS: an open-source robot operating system," in *IEEE International Conference on Robotics and Automation - Workshop on open source software*, vol. 3, no. 3.2, 2009, p. 5.

[31] Y. Tassa, T. Erez, and E. Todorov, "Synthesis and stabilization of complex behaviors through online trajectory optimization," in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012, pp. 4906–4913.

[32] V. Kumar, Z. Xu, and E. Todorov, "Fast, strong and compliant pneumatic actuation for dexterous tendon-driven hands," in *Proc. of the IEEE International Conference on Robotics and Automation*, 2013, pp. 1512–1519.