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*— In this work we present a research platform for fully autonomous commissioning tasks in intralogistics settings. The robot comprises a nonholonomic mobile base and a manipulation system consisting of a lightweight arm and an under-actuated gripper with active surfaces. The platform is capable of autonomously picking up pallets and loading them with unstructured goods in a manner which is safe for human workers sharing the environment. Target object handling is accomplished via a novel, redundant grasp representation which allows for redundancy in the gripper pose placement. This redundancy is exploited by an optimization-based control system which generates reactive manipulator motions on-the-fly without the delays occurring in sense-plan-act architectures.**

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.

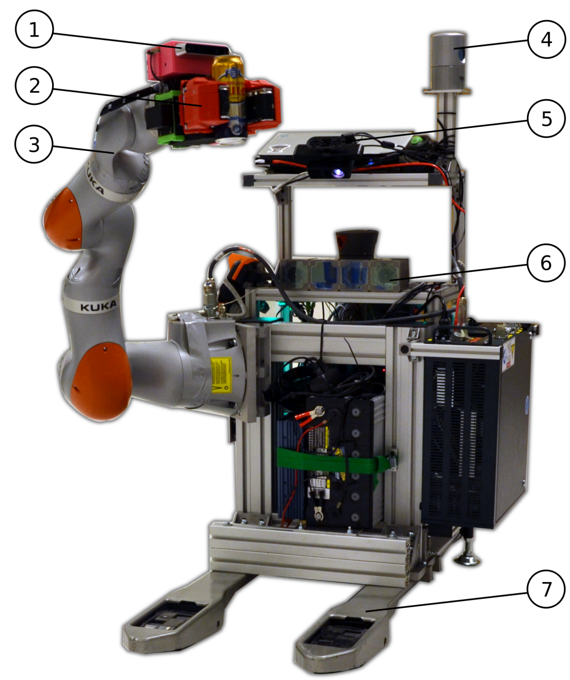


Fig. 1. *The APPLE platform:* A KUKA LBR iiwa arm (3) is mounted on a retrofitted Linde CitiTruck AGV (7). For localization, a Velodyne HDL-32 Lidar (4) is used, human worker detection is carried out with the RefleX camera system (6). In order to detect pallets, we employ a Asus Xtion Pro Live structured light camera (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 a Structure IO device (1) which is mounted on the gripper's palm.

The platform's mobile base consists of a non-holonomic Linde CitiTruck forklift AGV[1] which is able to detect and pick up pallets in designated loading zones. A KUKA LBR iiwa[2]. 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

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

[1] `http://www.citi-truck.com`
[2] `http://www.kuka-labs.com/en/service_robotics/lightweight_robotics/`

us to investigate various grasping/manipulation strategies.

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: below we outline the AGV navigation and motion planning scheme before presenting our solution for people detection in Section II-C. 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

The mobile platform is built upon a manual forklift "CiTi" from Linde Material Handling. The forklift is originally equipped with a motorized forks and drive wheel. The forklift has been retrofitted with a steering mechanism and a commercial AGV control system which is used to interface the original drive mechanism as well as the steering servo.

To assure safe operation the vehicle is equipped with a standard safety laser scanner (SICK S300) and an industrial prototype system (RefleX) for detecting workforce using reflective safety garments. To show the workers the intention of the vehicle, the intend path to be driven with the required occupied is projected onto the floor.

A. Challenges in Autonomous Navigation

The industry standard for autonomous navigation of forklifts is to use predefined trajectories where the trajectories are either manually defined or learned through teaching-by-demonstration from a human operator [4], [5]. Although conceptually simple, fixed trajectories limits the pallet handling to occur only at predefined fixed poses as well as simple strategies for handling unforeseen obstacles. The fundamental difficulties for motion planning of forklifts lies in the the non-holonomic constraints, the large sweep area it needs to occupy (due to its very non symmetrical footprint) while operating in limited work space.

To obtain reliable localization in large dynamic warehouses with high accuracy it is commonly used to mount reflectors in the environment and use a dedicated sensing device [6]. At this stage natural navigation has successfully been deployed in smaller entities where walls are commonly observed, however, larger and dynamic environments remains a challenge without additional infrastructure.

B. Navigation

The navigation modules ensures that the forklift is capable of moving save and autonomously through the work space environment to arbitrary load and unload poses with high accuracy. According to the AGV system provider Kollmorgen, the required end pose accuracy for picking up pallets is 0.03 m in position and 1 degree (0.017 radians) in orientation. The main component consist of trajectory generation, tacking controller and a localization system.

The trajectory generation on-line is done in two steps, where at first a kinematically feasible path with discretized start and goal poses are generated using a lattice planner [7]. This path is post-processed using a path smoother [8] which assure smooth collision-free continuous trajectory. The tracking of the trajectory is one using 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 [8].

The localization utilize a Velodyne HDL-32 3D laser scanner[3] which is used to construct a 3D map (using the 3D-NDT-OM map representation) of the static parts of the environment [9]. The map and odometry information is used to localize the vehicle in the presence of dynamic entities using a dual timescale approach [10].

To obtain the pose of the pallet to pickup the current system requires a rough estimate of the location of the pallet. In order to compute a final estimate based on sensory data from an Asus Xtion Pro Live[4] mounted on the AGV, a signed distance function (SDF) tracker [11] is used with a pre-defined SDF model of the pallet. The tracking is done while driving towards the provided initial pallet pose and the trajectory is recomputed on the fly, which depending on the pose offset may include a reverse operation.

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 [12]. 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. [13].

[3]http://www.velodynelidar.com/

[4]http://www.asus.com/Multimedia/Xtion_PRO_LIVE/

A. Challenges in Autonomous Grasping

In current autonomous grasping systems [14], [15], [16], grasp planning and manipulator motion planning are usually seen as independent sub-problems [17]. A database storing object models together with sets of pre-computed grasps is used to find suitable gripper poses/configurations [18], [19], [16]. In the online stage, sampling based planners [20] attempt to generate valid trajectories for the pre-planned grasps, which are executed in a feasible-first manner [14]. 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 [21], [22]. In this line of thought, opposed to define a set of discrete grasp poses, Gienger et al. [23], [24] 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 [25]. Lower-ranked tasks are then solved in the null-space of tasks with higher priority [21], [22]. A major step forward in this direction was achieved by Kanoun et al. [13], who extended the approach to include inequality tasks which significantly increases the expressiveness of the method. Recent advancements include the extension to dynamic control [26] and an efficient solution algorithm for the underlying optimization problem [27].

B. Object Detection

A typical commissioning scenario in a warehouse environment generally involves 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 this work, we assumed a simple scenario where objects of the same type are stored on a single pallet. As shown in Fig. 1, the gripper of the APPLE system is equipped with a Structure IO structured light depth sensor[5]. 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) [28]

[5] `http://structure.io/`

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

following the simple procedure summarized in Algorithm 1.

The basic idea of Algorithm 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 planning module described below.

C. Grasp Representation and Planning

Traditional data-driven grasp synthesis [29] 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

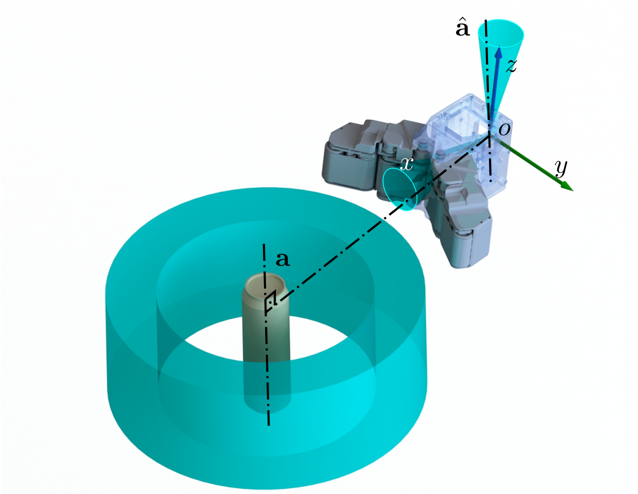


Fig. 2. *Grasp interval:* The shaded cyan regions illustrate the grasp interval for a cylindrical object. For a successful grasp, the palm frame origin $\boldsymbol{o}$ needs to lie inside the depicted cylindrical shell which is aligned with object axis $\boldsymbol{a}$. The cylinders height is limited 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 $\boldsymbol{o}$.

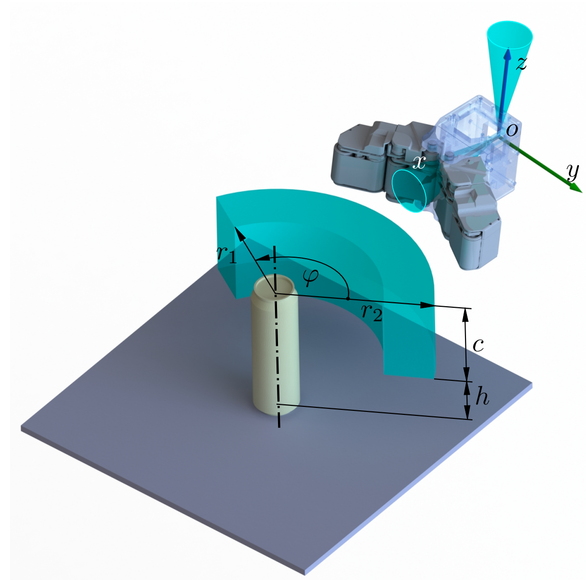


Fig. 3. *Truncated grasp interval:* During the online stage, the corresponding grasp interval shown in Fig. 2 needs to be truncated ($i.\,e.$, parameters for r_1, r_2, c, h and φ need to be determined) to accommodate the specific target object dimensions and to account for the fact that some regions of the grasp interval might not be feasible due to obstruction by the environment.

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 [23], [24], which are learned in simulation, we deliberately design grasp intervals to 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 [30]. 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 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.

During the grasp procedure, after the target object pose is detected as the mean and covariance of the cluster obtained from Algorithm 1, the grasp interval needs to be adapted to the specific scene and target object dimensions as illustrated in Fig. 3. For the evaluation presented in Section IV we pre-defined the corresponding parameters and gripper pre-grasp joint configuration, an appropriate programmatic approach is left to future work. One promising strategy currently being explored is to pre-compute a gripper collision map and use it for determining the valid grasping configurations around the target object. Once invalid grasping configurations have been pruned (by using the scene model), the remaining configurations can be clustered into continuous regions and used to determine the constraints in an automatic and collision-aware manner.

D. Manipulator Motion Generation

To leverage the freedom in pre-grasp position and orientation gained from the previously described grasp representation, we use a control-based approach for reactive, on-the-fly motion generation. More specific, the method of choice is the one developed by Kanoun et al. [13] which allows to formulate the grasp interval, as well as additional desiderata such as joint limit avoidance in form of task functions which subsequently are used to form stack of hierarchical tasks. In this paradigm, motions are generated instantaneously without the planning delays occurring in sense-plan-act architectures. In the following, we briefly revise the control concept. For an in-depth discussion the reader is referred to the original work presented in [13].

We lean on the notation in [27] 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 $\boldsymbol{p}(\boldsymbol{q})$. 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 $\dot{\boldsymbol{e}}^* = -\lambda \boldsymbol{e}$, 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

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

As stated in [27], this allows to transcribe lower bounds $\boldsymbol{J}\dot{\boldsymbol{q}} \geq \dot{\boldsymbol{e}}^*$, double bounds $\underline{\dot{\boldsymbol{e}}}^* \leq \boldsymbol{J}\dot{\boldsymbol{q}} \leq \bar{\dot{\boldsymbol{e}}}^*$ and equalities $\boldsymbol{J}\dot{\boldsymbol{q}} = \dot{\boldsymbol{e}}^*$ by reformulating the task respectively as $-\boldsymbol{J}\dot{\boldsymbol{q}} \leq -\dot{\boldsymbol{e}}^*$, $\begin{bmatrix}-\boldsymbol{J}\\ \boldsymbol{J}\end{bmatrix}\dot{\boldsymbol{q}} \leq \begin{bmatrix}-\underline{\dot{\boldsymbol{e}}}^*\\ \bar{\dot{\boldsymbol{e}}}^*\end{bmatrix}$ 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 in (2) 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 one 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 $p = 1, \ldots, P$

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

The final joint velocity vector $\boldsymbol{q}^*$ is then obtained from the P^{th} solution of (4). Opposed to sampling based planners, the chosen control scheme allows to incorporate qualitative requirements ($e.g.$, the desired gripper alignment in the presented case) during motion generation and allows for redundancy exploitation via appropriate task function formulations.

Manipulator 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. To this end, the APPLE platform exploits the compliant low-level control schemes and contact detection abilities of the chosen manipulator.

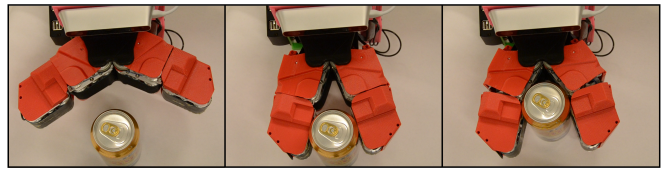
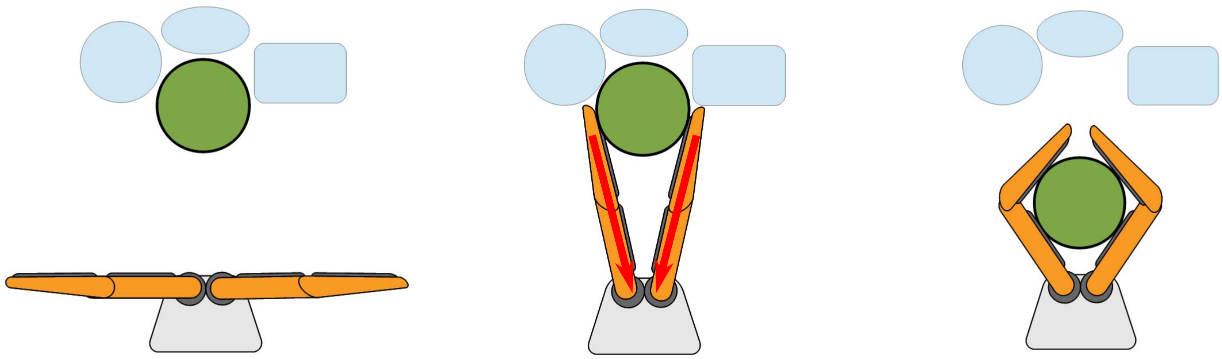


Fig. 4. *Pull-in grasping strategy:* Depicted is a sequence of intermediate grasp states where the belts of the gripper are used to pull the object towards its palm which results in a transition from a fingertip to an enveloping grasp.

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. 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 [16] showed, that in cluttered scenes fingertip grasps are more likely to be feasible 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. 5. As demonstrated previously [16], this strategy is especially effective in achieving stable grasps, starting from a relatively wide range of initial gripper poses with respect to the target object.

The grasping controller was implemented by means of low-level current controllers acting on the gripper open/close DoF and the actuated belts respectively. Current control of the motors emulates a force/torque controller, since the motor's current absorption increases with increasing effort on the output and thus with increasing grasping force. The pull-in strategy is implemented in three steps. First, a relatively low current is set to the open/close motor and the fingers start closing. Once the fingers come into contact with the object and the target current is reached, the gripper blocks and there is no more motion. We detect this situation by monitoring the encoder on the open/close motor and enter the second phase of the grasping process: namely, we actuate the belts to pull in the target object. We then monitor the encoders on the belts and on the phalanges of the fingers. If the belts block and the phalanges have wrapped around the object, we have achieved an enveloping grasp and enter the last stage of the process: we set a higher current to the motor in order to ensure a firm grasp. The main parameters to this routine (the current thresholds for contact and final enveloping grasp) depend on the properties of the object — namely the friction coefficient and weight. In this work, the parameters were experimentally tuned and good values were found for a set of common household objects.



(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).

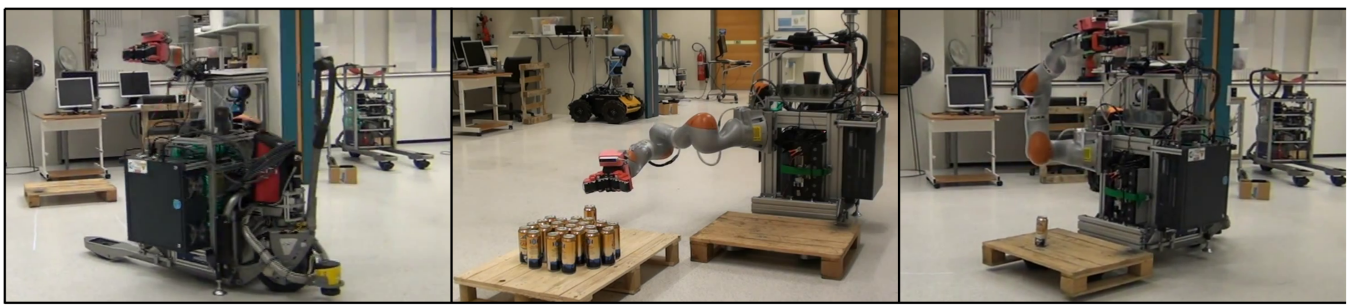


Fig. 6. *Evaluation setup:* (Left) the robot picks up an empty pallet in a designated zone; (Middle) the robot navigates to a loading zone where a can is detected and picked up; (Right) the loaded pallet is transported to a drop-off location.

IV. Evaluation

For an early evaluation of the system, we set up a simplified commissioning scenario as depicted in Fig. 6. Tho this end, the previously described software components were implemented in the Robot Operating System (ROS) [31] framework. For manipulator motion control (see Section III-D) we use the task function formulations in [32] for joint limit avoidance, obstacle avoidance and to formulate the inequality constraints illustrated in Fig. 3 in order for the manipulator to reach the grasp interval. An off-the-shelf solver [33] is used to carry out the optimizations for the motion control according to (4). The run time for the whole procedure is approximately 4 minutes of which 2 minutes are spent on object detection and grasping (see the attached video submission).

V. Discussion and Outlook

In this paper, we presented a research platform for autonomous picking and palletizing (APPLE) for fully autonomous commissioning tasks in intralogistics settings. The APPLE robot comprises a nonholonomic mobile base with the ability to autonomously detect and pick standard EUR half-pallets from designated loading areas. Also incorporated is a camera system for detection and avoidance of human workers wearing reflective vests. The manipulation system for loading/unloading unstructured goods from pallets operates on a novel redundant grasp representation as intervals in task space which allows to incorporate empirical knowledge. We leverage the obtained redundancy by generating reactive manipulator motions on the fly using a prioritized control approach which allows to formulate the target object picking as a stack of hierarchical tasks [13]. We provide an early experimental evaluation of the APPLE system by means of a simplified commissioning task (see the video attachment to this article).

The presented work is limited to the picking of objects with cylindrical shapes, future work will aim at extending our grasp interval representation to other primitive shape types. Also, we work on a computational method to parametrize the interval for a current scene which, right now, is done empirically.

Acknowledgments

The authors would like to thank Per Sporrong, Bo-Lennart Silfverdal, Bengt Åsberg and Joakim Larsson at Örebro University for their engineering support.

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. Hellstrom and O. Ringdahl, "Follow the past: a path-tracking algorithm for autonomous vehicles," *Int. Journal of Vehicle Autonomous Systems*, vol. 4, no. 2, pp. 216–224, 2006.

[5] J. Marshall, T. Barfoot, and J. Larsson, "Autonomous underground tramming for center-articulated vehicles," *Journal of Field Robotics*, vol. 25, no. 6-7, pp. 400–421, 2008.

[6] K. Hyyppa, "Method of navigating an automated guided vehicle," Mar. 7 1989, uS Patent 4,811,228.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] D. R. Canelhas, T. Stoyanov, and A. J. Lilienthal, "Sdf tracker: A parallel algorithm for on-line pose estimation and scene reconstruction from depth images." in *IROS*. IEEE, 2013, pp. 3671–3676.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

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

[18] 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.

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

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

[21] 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.

[22] 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.

[23] 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.

[24] 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.

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

[26] 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.

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

[28] R. B. Rusu and S. Cousins, "3D is here: Point Cloud Library (PCL)," in *Proc. of the IEEE International Conference on Robotics and Automation*, 2011, pp. 1–4.

[29] 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.

[30] 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.

[31] 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.

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

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