This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2018.2800744, IEEE Transactions on Industrial Informatics

Integrating Different Levels of Automation: Lessons from Winning the Amazon Robotics Challenge 2016

Carlos Hernández, Mukunda Bharatheesha, Jeff van Egmond, Jihong Ju and Martijn Wisse

***Abstract*—This article describes Team Delft's robot winning the Amazon Robotics Challenge 2016. The competition involves automating pick and place operations in semi-structured environments, specifically the shelves in an Amazon warehouse. Team Delft's entry demonstrated that current robot technology can already address most of the challenges in product handling: object recognition, grasping, motion, or task planning; under broad yet bounded conditions. The system combines an industrial robot arm, 3D cameras and a custom gripper. The robot's software is based on the Robot Operating System to implement solutions based on deep learning and other state-of-the-art artificial intelligence techniques, and to integrate them with off-the-shelf components. From the experience developing the robotic system it was concluded that: 1) the specific task conditions should guide the selection of the solution for each capability required, 2) understanding the characteristics of the individual solutions and the assumptions they embed is critical to integrate a performing system from them, and 3) this characterization can be based on 'levels of robot automation'. This paper proposes automation levels based on the usage of information at design or runtime to drive the robot's behaviour, and uses them to discuss Team Delft's design solution and the lessons learned from this robot development experience.**

***Index Terms*—grasping, manipulators, motion planning, object recognition, robot control**

I. INTRODUCTION

THE Amazon Robotic Challenge (ARC) [1], [2], was launched by Amazon Robotics in 2015 to promote research into unstructured warehouse automation and specifically robotic manipulation for picking and stocking of products.

Low volume, high-mix productions require flexibility to cope with an unstructured environment, and adaptability to quickly and cost-effectively reconfigure the system to different tasks. Current commercial solutions have mainly focused on automating the transport inside the warehouse, whereas only few solutions exist for the individual handling of the products [3], and are usually limited to one product type at a time[1]. Currently there is a diversity of grippers available such as 2-finger grippers [4], VERSABALL [5], or more advanced robotic hands such as [6] or [7] that can be customized for different applications. The selection of the gripper for a manipulation applicationgreatly affects the flexibility and requirements of the grasping process. More flexible robotics solutions are needed that benefit from advances in artificial intelligence and integrate them with these more dexterous and reliable mechanical designs for grippers and manipulators.

The integration of these robot technologies into an agile and robust solution, capable of performing on the factory floor, is itself an engineering challenge [8]. During a robotic application development design decisions need to be made, e.g. about feedback control vs. planning, that entail trade-offs between flexibility and performance. For example, in the first ARC edition in 2015, the winning robot used a feedback approach with visual servoing, achieving a robust pick execution that outperformed the competitors. However, the public media was disappointed about the general speed performance of the robots [9]. The average pick time for the winner was above one minute ($\sim$30 sorts per hour), while industry demands the $\sim$400 sorts/h achieved by humans [2].

There were two key ideas guiding Team Delft's approach to building the ARC 2016 robot: 1) reuse available solutions whenever possible, and 2) chose them so as to automate the system to the level required by the different challenges in the competition, making useful assumptions based on the structure present in the task.

To reuse available off-the-shelf solutions, Team Delft robot was based on an industrial manipulator and 3D cameras, and the robot software was based on the Robot Operating System (ROS) [10]. ROS provides tools and infrastructure to develop complex robotic systems, runtime communications middleware, and the ROS open-source community provides off-the-shelf components for many robot capabilities.

There is a variety of aspects that have been identified useful to characterize robotic systems [11]: modularity vs. integration, computation vs. embodiment, planning vs. feedback, or generality vs. assumptions. The dichotomy *planning vs. feedback* in robotics represents only two (important) classes in the spectrum of solutions. These range from open-loop solutions that exploit assumptions and knowledge about the task and the workspace at design time, to feedback strategies that use runtime information to drive robot's behaviour and deal with

Manuscript received May 31, 2017; revised August 26 and November 12, 2017; accepted January 21, 2018.

*All authors gratefully acknowledge the financial support by the European Union's Seventh Framework Programme project Factory-in-a-day (FP7-609206). We are very grateful to all our team members who helped create the winning robot, and supporting sponsors and colleagues. The authors would like to thank RoboValley, the ROS-Industrial Consortium, Yaskawa, IDS, Phaer, Ikbenstil and Induvac, the people at the Delft Center for Systems and Control, TU Delft Logistics, and Lacquey B.V. Special thanks to Gijs van der Hoorn for his help during the development of the robotic system.

The authors are with TU Delft Robotics Institute, Delft University of Technology, Mekelweg 2, 2628 CD Delft, The Netherlands (e-mail: {c.h.corbato, m.bharatheesha, m.wisse}@tudelft.nl, daniel.jihong.ju@gmail.com, jeff.vegmond@gmail.com)

Digital Object Identifier (DOI): see top of this page.

[1]E.g. see work of Profactor GmbH. at: https://www.profactor.at/en/solutions/flexible-robotic/handling/

uncertainties. After analysis and reflection on the ARC robot development experience, different *levels of robot automation* are proposed in this paper to characterize the design solutions. In Team Delft's robot design, the different solutions were chosen to automate every part of the robotic system to the level required. Different automation solutions render different system properties in terms of flexibility, performance, and adaptability.

Section II discusses the requirements posed by the ARC 2016 competition scenario, and analyses the challenges it poses to robot perception, manipulation and task planning. In section III the levels of robot automation are presented, and used to explain Team Delft's robot concept in section IV. The performance of the system is discussed in view of the levels of automation in section V, and some lessons learned are reported. Finally section VI provides concluding remarks.

II. Manipulation in the Amazon Robotics Challenge

The Amazon Robotics Challenge (ARC) stems from a broader and fundamental research field of robotic manipulation in unstructured environments. The two tasks for the 2016 challenge [12] involved manipulating diverse, small sized products to pick and place them from an Amazon shelving unit (*the shelf*) structured in twelve bins, to a temporary container (*the tote*), as is illustrated in Fig. 1. We begin this section by providing further technical details of the challenge, followed by a comparative analysis of the challenge to relevant scientific problems.

A. The Amazon Robotics Challenge 2016

The challenge for the year 2016 was titled Amazon Picking Challenge and consisted of two tasks to be autonomously performed by a robotic system:

The Picking Task consisted of moving 12 products from a partially filled shelf, into the tote. Some target products could be partially occluded or in contact with other products, but no product would be fully occluded. Each of the 12 bins contained exactly one target product as well as any number of non-target products and every target product is only present in a single bin. The tote is initially empty in this task.

The Stowing Task was the inverse to the Pick Task: moving the contents of the tote (12 products) into the bins of the shelf, which already contain some products. The products in the tote could be partially or completely occluded below other products. There was no target location for the products in the tote, but different score for stowing them into more cluttered bins. No restrictions were given on how to place the products in the shelf bins, apart from not damaging them or the shelf and not protruding more than 1cm.

In both tasks the robot had 15 minutes to fulfil the order, which was specified by a task file, and report the final location of all the products in an analogous output file. The task file contained information of what products were located in which bin or tote and it identified the target products. The task file did not contain information about the physical location of products within their respective container. The target products could be handled in any order and all the product could be moved to any bin, as long as the final contents of each bin and the tote were correctly reported in the output file. The performance of the robot was evaluated by giving points for correctly placed items and subtracting penalty points for dropping, damaging or misplacing items (i.e. incorrectly reporting its location in the output file). The amount of points for a specific operation would depend on the difficulty of the object and the cluttering of the bin. The time to accomplish the first successful operation would be the tiebreaker.

Fig. 1. The products in the Amazon Picking Challenge 2016 in the tote and in the bins of the shelf, from [12].

B. Manipulation in unstructured environments

The ARC scenario is representative of the challenges in handling applications in a warehouse or the factory floor. The robot has to perform a few simple tasks in a closed environment, but it is only semi-structured. Unlike dynamic, open environments where autonomous robots have to cope with unbounded levels of uncertainty, here it is limited. However, uncertainty is still present, in the target products characteristics, their position and orientation, and the workspace conditions.

The set of 39 product types used in the competition includes books, cubic boxes, clothing, soft objects, and irregularly shaped objects. They were chosen to be representative of the products handled on a daily basis at an Amazon warehouse. They presented realistic challenges for perception, grasping and manipulation: reflective packaging, wide range of dimensions, and weight or deformable shapes.

The products are stored mixed in any position and orientation inside the shelf's bins, partially occluding each other, sometimes placed at the back. Bins could be too cluttered even for a human hand to easily pick the target item. The shelf construction with metal parts and cardboard divisions resulted in wide tolerances and asymmetries. Besides, the narrow opening of the bins (21 cm x 28 cm) compared to their depth (43 cm) limited the manoeuvrability inside, and caused difficult dark lighting conditions. The highly reflective metal floor of the bins contributed to the challenges for any vision system. In addition, the position of the entire shelf had +/-3 cm tolerance.

The variety of shapes, sizes and weights of the objects also posed an interesting challenge for object manipulation.

This variety entailed studying and applying different grasp synthesis methods such as pinch grasping [13], [14] and suction grasping, successfully used in the previous edition of ARC [11]. The limited space for manipulation discarded cage grasping strategies. Regarding grasp synthesising, despite the extended literature on grasping of unknown objects [15], the fact that the products were known well in advance made fine-tunned heuristics promise much better performance, as early tests demonstrated.

III. Levels of automation

The performance and flexibility of a robotic application depends on the assumptions and design decisions made to address uncertainty. A proper understanding of these decisions is specially important in robots that perform more traditional automation tasks, but with challenging flexibility in not so-well structured environments, such as the ARC. For this a characterization of the solutions in *levels of robot automation* is proposed, based on the experience gained developing Team Delft's robot. Our model is inspired by that of Parasuraman *et al.* [16]. While that framework supports decisions about which functions to automate, and to what extend, with a focus on the human interaction factor, the model presented here applies to the case of full automation of a function. It provides a basis for deciding how to automate those functions, in view of the uncertainty present and the reliability required. The main criteria to differentiate automation solutions is the timing of the information used to drive the behaviour of the system. Assumptions are prior information that is used at design time to determine a certain behaviour, reducing the flexibility of the system, but generally optimizing its performance. On the other hand, closed control loops in a robotic system use runtime information to adapt the system behaviour to the actual circumstances on-the-fly.

In traditional automation, the environment and the task are fixed and assumed perfectly modelled. This allows to fix at design time the sequence of operations and open-loop robot motions. Uncertainty is reduced to minor allowed deviations on product placement and geometry, which are accommodated for by robust and compliant hardware designs. This *'level 0'* automation allows to maximize the motion's speed leading to very high performance. However, it has no flexibility: robot's behaviour is fixed during design, no runtime information is used to adapt to deviations.

Open-loop automation solutions typically include error handling mechanisms, so that the robotic system can accommodate for foreseeable events during its design. These *'level 1'* solutions introduce sensing capabilities in the system to verify *a posteriori* the result of actions. For example in suction-based object handling the pressure in the suction cup can be checked to confirm a successful grasp or to detect dropping the object.

In *'level 2'* of robot automation, more advanced and rich perception is used to drive the robot behaviour at runtime, following the so called *sense-plan-act* paradigm. The complete sequence of control actions is computed based on a predefined model of the world and initial run-time sensor information that accommodates any run-time uncertainty. A typical example is a vision-based solution that locates target objects. The limitations of this approach are well known in robotics and artificial intelligence fields [17].

In feedback control (*'level 3'*), action is dynamically computed at a certain frequency using runtime sensor information. Often, the target variables cannot been sensed at the desired frequency, or they are not directly accessible at all. In these cases, an estimation is used to close a feedback loop at runtime. The controller of a robot manipulator, closing a control loop for its joint state, is an example of *'level'* present in Team Delft robot.

Finally, a *'level 4'* solution uses predictions in addition to the current sensor information to optimize its response to an uncertain environment. This is the case in systems that use any flavor of model predictive control [18], in which a more or less complex model of the system dynamics is used to optimize the control action based on the predicted evolution.

The selection of the level of automation for each specific problem in a robot manipulation application implies a trade-off between flexibility, performance, and resources. In the following sections the Team Delft robot for the ARC 2016 is discussed, explaining the rationale for the different technological approaches chosen following the model of *'levels of automation'*.

IV. Robotic System Overview

Based on the analysis of previous experiences in the ARC [11], [2], Team Delft's solution targeted three key performance objectives to maximize scoring in the competition: be able to complete all the tasks, robustness and speed. The design approach to address them was to develop the robot automation level more efficient considering the uncertainty challenges in the tasks, and to reuse existing hardware and software components.

A. System Requirements

The performance objectives were decomposed into specific system requirements for robot manipulation. Completing the picking and stowing tasks requires the robot to handle all the products in any position in the shelf and the tote. This entails the following requirements:

Req. 1: to recognize and locate any of the products in any place inside the shelf or the tote.
Req. 2: to reach any location inside the bins with enough manoeuvrability.
Req. 3: to achieve and hold a firm grasp on all different products.

Robustness is a must in real-world applications that need to perform with almost no downtime. In the competition only one attempt[2] was allowed for each task, so any failure leading to the robot stopping or crashing is fatal. Speed is also critical for production systems. In Team Delft's strategy, speed allows the robot to perform several attempts to pick a target difficult to

[2] A reset was allowed: the task could be restarted from the beginning but with a penalty [12].

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2018.2800744, IEEE Transactions on Industrial Informatics

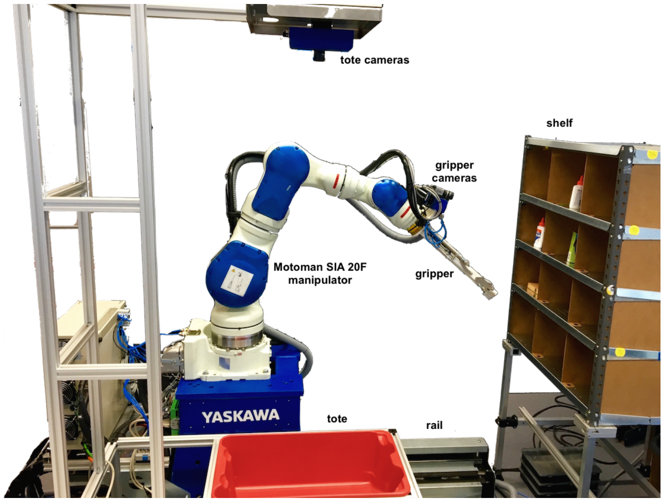

Fig. 2. Team Delft robot setup in the APC workcell.

grasp, and also move other objects for clearance, during the 15 minutes allowed for each task. This simplifies the manipulation actions needed, leading to a more robust system.

B. Robot Concept

Team Delft's robotic system is shown in Fig. 2. It is based on an industrial manipulator mounting a 3D camera to scan the contents of the shelf's bins and a custom, hybrid gripper featuring a suction cup and a pinch mechanism. An additional fixed camera allows scanning the tote contents. The selection of this hardware will be justified together the explanation of the main robot functionality each device supports, in subsections IV-C, IV-D and IV-E.

The ARC competition requires the robot to operate autonomously to complete tasks defined in a computer file that defines the current inventory of the shelf and the tote, and, for the pick task, the target product in each bin to be placed in the tote. Team Delft's solution for the picking and the stowing tasks is to decompose them into a plan of pick&place operations that is sequentially executed by the robot arm.

1) Task Planning: For the Picking Task, the picking order of each target is computed to maximize scoring and minimize risk, considering i) the points for each product, and ii) system's confidence to handle each product, from experimental results, and iii) the need to move occluding objects (see rightmost flow in Fig. 3). This way the plan of pick&place operation for all targets in the task is created. The plan is updated at the end of each operation according to its success or any fallback triggered (see failures in the right side of Fig. 3), as will be explained in section IV-F. For the stowing task, a simple heuristic selects as a target the detected product that is closer to the tote opening, since all the contents in the tote have to be stowed.

2) Pick&place: The pick&place operations required to handle the products in the competition have a fixed structure in a closed, semi-structured environment: pick target X that is located in bin Y or the tote, and place it on the tote or bin Y'. Therefore a *'level 2'* robot control solution was designed, consisting of a sequence of actions that follows the sense-plan-act paradigm. The general steps and the main actions depicted in Fig. 3 are as follows:

Sense
The system uses the 3D camera information of the target's container (bin or tote) to: i) detect and estimate the 6D pose of the item, and ii) obtain collision information of the container to later plan the motions, in the form of a filtered Pointcloud of the cluttering of the container. In the Pick Task scenario, the robot has to previosuly move to obtain the camera information for the target bin. Additionally, the actual position of the bin is also estimated, for a more detailed collision model of the environment. In the Stow task, a Pointcloud model of the tote is used, since its pose is perfectly known.

Plan
Using the estimated pose of the target item and its known characteristics, the system computes the grasping strategy and a grasp candidate (a pose for the gripper to pick the object). The sensed Pointcloud information is integrated in an octomap with the known environment geometry, stored in the Universal Robot Description Format (URDF) to generate a collision-free plan to approach the grasp candidate pose, pick the product and retreat from the shelf.

Act
The previous motion plan is executed as a feedforward action, including gripper configuration and activation on the way to pick the item. Pick success is confirmed with the pressure in the suction cup (for suction-based grasps). If so, the robot moves to drop the item in the tote using offline generated trajectories.

Thanks to the structure of the environment, to place the products a robot automation *'level 0'* solution was designed that uses pre-defined motions to drop them either in the tote or the shelf's bins in a safe manner to comply with the competition rules. In the case of placing the items in the tote, it is divided in 6 predefined drop locations, and the task planner logic makes sure that: i) no heavy products are dropped were fragile items have been places and ii) no more than 3 products are dropped in the same location, so that the objects do not protrude from the tote. In the case of moving occluding items to another bin, the task planner logic selects the least cluttered bin from those that no longer need to be accessed (to keep the environment static). The robot moves to a fixed location in that bin, making use of the assumption that thanks to gravity any cluttering is in the lower part, and any standing items will be toppled inwards.

Sections IV-C to IV-E describe the solutions designed for all the robot capabilities required for the previous actions, grouped into object detection and pose estimation, grasping and robot motion, including the main hardware and software components involved.

C. Vision-based Perception

To address Req. 1, the robot needs to recognize and locate the objects captured by the camera, knowing what the object is and where it locates in the image.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2018.2800744, IEEE Transactions on Industrial Informatics

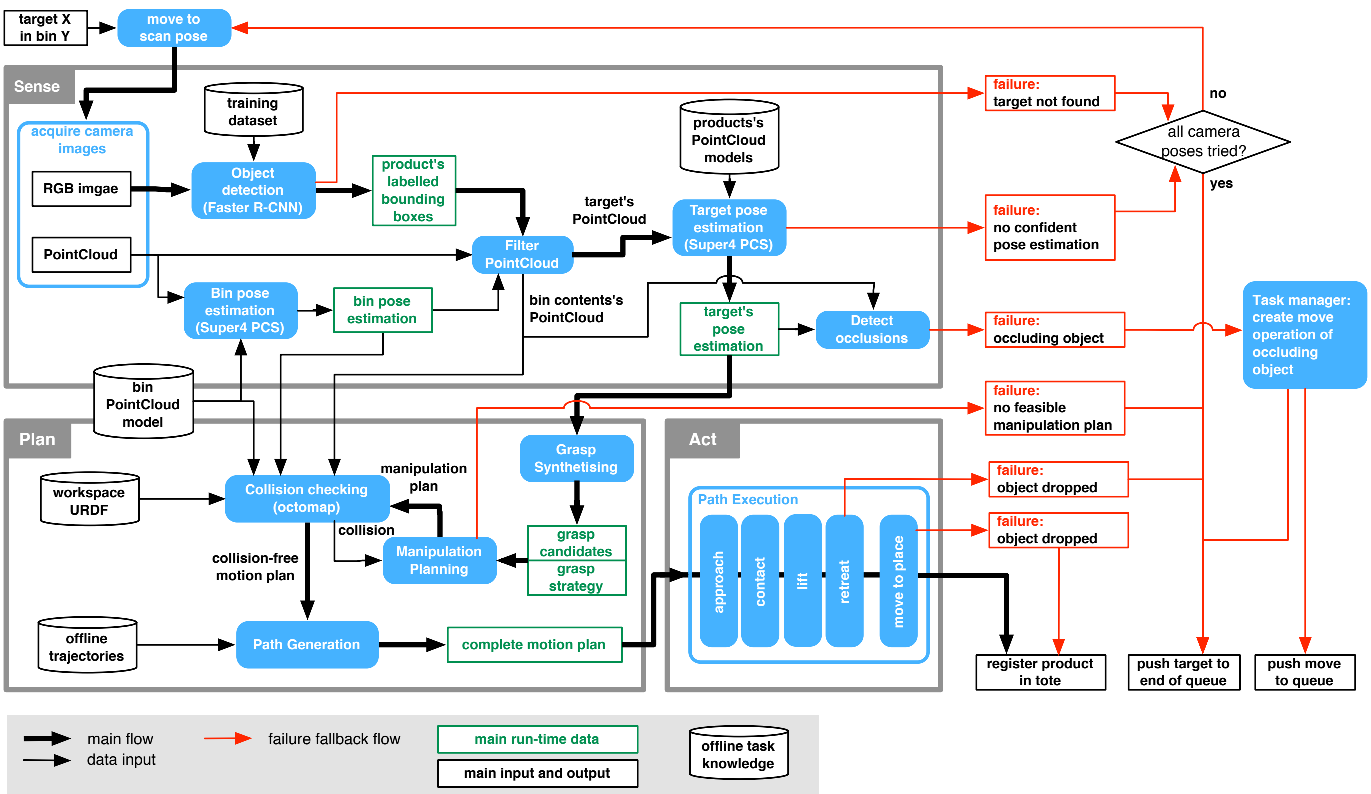

Fig. 3. Schema of Team Delft's sense-plan-act workflow for picking a product *X* from the shelf's bin *Y*. The sense step consists of: a) detection of the target item and estimation of its 6d pose, and b) obtain collision information inside the bin, in the form of a PointCloud.

1) Cameras: To scan the bin an industrial camera system is mounted on the gripper. It includes a Ensenso N35 3D camera that provides low noise Pointcloud data, and an IDS UI-5240CP-C-HQ high-definition camera that provides RGB images. An array of LEDs improves robustness to lighting conditions. A similar system is fixed on a pole to scan the tote.

2) Object Detection: The object detection module takes RGB images as input and returns a list of the object proposals. Each proposal contains the label and the location, determined by a bounding box enclosing the object. The proposals are ranked in descending order based on the confidences, varying from 0 to 1. The proposal with the highest confidence for the expected item was counted as the detected object.

One of the main difficulties for object detection is that each instance varies significantly regarding size, shape, appearance, poses, etc. The object detection module should be able to recognize and locate the objects regardless of how objects from one category differ visually. A model that has high capacity and can learn from large-scale data is required. Deep Neural Networks are renowned for it high capacity, especially the Convolution Neural Networks (CNN) have recently shown its ability to learn large-scale data efficiently, improving the benchmark performance of large scale visual recognition problems significantly since 2012 [19], [20].

Girshick et. al. [21] adopted the Convolutional Networks for classification, and selective search [22] for region proposal, in their region-based Convolutional Networks (R-CNN) framework, achieving a performance improvement by a large margin. One of the limitations of their work is that it took about 47 seconds[3] to create the region proposals and predict the object categories, for each image. Following studies [23], [24] accelerated the processing cycle time to 198 milliseconds by applying the CNN more efficiently, including extracting convolutional features for the whole image and sharing CNN for region proposal and classification. The resulting method is referred to as Faster R-CNN[24], [20].

The significant processing speed acceleration of the Faster R-CNN consolidates the basis of nearly real-time object detection in robotic applications. This is the reason Faster R-CNN was adopted in Team Delft's solution to detect objects in both the shelf and the tote of the ARC.

Training a Faster R-CNN model requires a ground truth data set, in this case RGB images annotated with the bounding boxes and labels of detected objects. In the ARC setup, objects were placed in two different scenes, either in a dark shelf bin or a red tote. Therefore, two different models were trained to detect objects in the two different scenes. A total of three sets of RGB labeled images, were used for training:

Base Images of all the products were recorded automatically. Objects were put on a rotating plate against a monochrome background, and a camera was attached to a robot arm, taking images from different angles. Annotations were generated after automated object segmentation by thresholding. Images were aug-

[3]All process timings run on one Nvidia K40 GPU (graphics processing unit) overclocked to 875 MHz as provided in papers [23], [24].

TABLE I
EVALUATION OF THE CONVOLUTIONAL NEURAL NETWORKS

Network	Bin test mAP	Tote test mAP
Base Model	16.1%	7.9%
Bin Model (bin data only)	82.9%	-
Bin Model	85.7%	-
Tote Model (tote data only)	-	90.0%
Tote Model	-	92.5%

mented by replacing the monochrome background with random pictures after creating labels. This set contains in total 20K images.

Bin Images were taken for objects randomly placed in the bin. Annotations were created manually. This set includes 672 images.

Tote Images were taken for objects randomly placed in the tote. Annotations were created manually. This set contains 459 images.

The pre-trained weights of the VGG net [25] were used as initialization for the convolutional filters while the other filters were initialized with small random values drawn from a Gaussian distribution [26].

Given the three different sets of labeled images, five models were trained in a two-step strategy:

Step 1 Trained the initialized model with all the images from the Base set, obtaining a *Base model*.

Step 2 Fine-tuning the Base model with scene-specific images, the Bin set and the Tote set, obtaining scene-specific models, a *Bin model* and a *Tote model*.

The first model is the Base model. For both the Bin and Tote models, two different models were trained. One model uses only the data from the respective environment (omitting step 1 of the two-step strategy), whereas the second model is obtained by refining the Base model with the environment specific training set; applying both steps.

The trained models were tested using 10% of the Bin set, and Tote set, respectively, as two test sets. The test sets were excluded from the training procedure. An object proposal was counted as correct if it had more than 50% of the area overlapped with the corresponding annotation. Average Precision (AP) were used to evaluate the ranked list of object proposals for each item category, and the mean over the 39 categories, known as Mean Average Precision (mAP), were used as the performance measure of the models. The Mean Average Precision varies from 0 to 1, and higher mAP indicates that the predictions match better with the annotations.

The result of this evaluation can be seen in table I. From this it is observed that the best results are obtained by refining the generic Base model with environment specific data. The Bin model was used in the ARC 2016 for the picking task and the Tote model was used for the stowing task.

3) Object pose estimation: While object detection localizes objects in 2D, handling the target objects requires knowing the 3D pose of the object with respect to the robot. The chosen approach separates pose estimation in two stages: global pose estimation and a local refinement step.

Global pose estimation was done using Super 4PCS [27]. Since this method compares a small subset of both a model Pointcloud and the measured Pointcloud for congruency, it can obtain a good global estimation of the object pose. This global estimation is then used as an initial guess in applying Iterative Closest Point (ICP) [28] for a close approximation.

For these methods, Pointclouds without color information were used. While it has been suggested [27] that using color information is possible in Super 4PCS, no analysis of its effect on the pose estimation performance was reported in that study. Furthermore it would have required obtaining accurate colored Pointcloud models of the objects, while for most objects a simple primitive shape can be used to generate a Pointcloud model if color information is ignored. For some more elaborately shaped objects (a pair of dog bones and a utility brush for instance), a 3D scan without color information has been used as a model.

It should be noted that the Super 4PCS method inherently uses the 3D structure to obtain a pose estimation. Lack of such structure in the observed Pointcloud leads to suboptimal results. For example, observing only one face of a cuboid object could lead to the wrong face of the model being matched to the observation.

D. Grasping

Team Delft's approach to grasping and manipulation was to simplify the problem to a minimum set of action primitives, relying on the following additional requirements:

Req. 4: a suitable grasp surface is always directly accessible from the bin opening that allows to grasp and retreat holding the product, and no complex manipulation inside the bin or the tote is needed. This way the *'level 2'* assumption of environment invariance holds.

Req. 5: the system should be able to find a collision-free path to grasp the product, and a retreat path holding the product.

Req. 6: the gripper is able to pick and robustly hold any of the 39 product types, compensating for small deviations, minor collisions of the held product, inertia and gravity effects on grasp stability.

1) Gripper: A custom hybrid gripper was tailored to handle all items in the competition (Req. 6). It includes a suction cup based on low vacuum and high volume for robustness, and a pinch mechanism for the two products difficult to grasp with suction: a 3 pound dumbbell and a pencil holder made out of wire mesh. The gripper's 40cm length allows to reach all the way inside the bins without the bulky robot wrist entering, and its lean design facilitates manoeuvrability inside the reduced space. Bulkines and resulting limited maneovrability inside the bins was also the reason why robot hands were discarded. The suction cup features a 90° rotation to provide an extra degree of freedom. This allows using the front surface of the object facing the robot for grasping, facilitating Req. 4.

2) Grasp Strategies: The grasping strategy is chosen at run-time based on the type of product, its pose and the surrounding cluttering, from these primitives: *front suction*, *side or top suction*, and *pinch*. The chosen primitive is parametrized to

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2018.2800744, IEEE Transactions on Industrial Informatics

(a) Top suction (b) Front suction (c) Side suction (d) Pinch

Fig. 4. The different grasp strategies possible with Team Delft's custom gripper.

the target product by computing the grasp candidate pose and an associated manipulation plan, using a priori knowledge and runtime information.

3) Grasp Synthesising: The grasp candidate is a pose that if reached by its end-effector allows the robot to grasp the product activating the gripper according to the chosen primitive (by generating suction or the pinch mechanism). For non-deformable products the grasp candidate is generated using heuristics that store grasp possibilities for the different product types, based on geometric primitives and the structure of the workspace, as detailed in [29]. Since a 3D pose estimation is not possible for deformable products, grasp candidates are obtained using the surface normals of the detected object's Pointcloud, and ranked according to their distance to its centroid.

E. Robot Motion

The motion module is responsible for moving the end-effector to all the poses needed along the sense-plan-act behaviour, fulfilling Req. 2 for reachability, Req. 5 for grasping and the requirement for speed.

1) Robot: To choose a robot manipulator that could execute all required motions, a workspace reachability analysis using the MoveIt! [30] was conducted. The robotic system designed consists of a 7 degrees of freedom SIA20F Motoman industrial manipulator mounted on a rail perpendicular to the shelf's front. The resulting 8 degrees of freedom allows to reach all the bins with enough manoeuverability.

The motion problem was simplified using two assumptions about the workspace uncertainty:

1) outside the shelf the environment is static and known, and the task involves a finite set of poses to scan the bins and the tote, and to access them, so motions can be pre-planned offline (*'level 0'* solution);
2) inside the shelf and the tote the environment is also static but unknown. However, it has some structure due to: the walls, the given condition of products not laying on top of each other, gravity, and in the case of the shelf the assumption of one surface accessible from the bin opening.

2) Motions outside the shelf and tote: Using assumption 1) a *'level 0'* solution was implemented to implement the motions needed outside the shelf and tote. Around 30 end-effector poses were pre-defined, and collision-free trajectories between all of them were planned offline.

3) Manipulation Planning: Using the second assumption, the manipulation strategy was designed from a motion perspective as a combination of linear segments to *approach*, *contact*, *lift* and *retreat*. These segments are computed online from the grasp candidate poses using cartesian planning. Collisions are accounted for using the shelf's or tote's 3D model and online information of the surroundings by generating an occupancy octomap from the scanned Pointcloud.

4) Path Generation and Execution: Finally, the offline trajectories and the manipulation segments are stitched into a complete time parameterized motion plan. This process optimizes the timely execution of the motions. It allows for custom velocity scaling to adapt the motions to safely handle the heavier products. This process also synchronizes along the trajectory the timely configuration (to the desired strategy), and later activation of the gripper to pick the target product, and the verification of the pressure sensor in the gripper after retreat. Finally, the resulting trajectory is executed by the robot manipulator, also controlling the gripper.

F. Failure management

Special focus was given to the overall reliability of the robotic system. The system can detect a set of failures during the sense, plan and act phases and trigger fallbacks to prevent a stall. For example, if the target product cannot be located, or estimate its pose, different camera poses are tried. If the problem persists it will postpone that target and move to the next one. A failed suction grasp is detected by checking the vacuum sealing after execution of the complete grasp and retreat action. In that case, it is assumed that the item dropped inside the bin and retries the pick later. If the vacuum seal is broken during the placement in the tote, the item is reported to be in the tote, since it can actually be the case, and there is no gain for reporting dropped items. For a pinch grasp, the system could only validate the complete pick and place operation by checking the presence of the picked item in the image from the tote.

V. Discussion

The Amazon Robotics Challenge is a valuable benchmark for robot manipulation solutions. It provides interesting indicators to measure the advantages and limitations of the different robotic solutions. Following we discuss the performance of Team Delft robot using the automation levels framework presented in section III and analysing the different trade-offs of using run-time feedback or design assumptions to address the uncertain conditions.

A. Evaluation

Table II summarises the results of Team Delft's robot in the competition to win both challenge tasks [29]. The best scoring teams in the table shared some features, e.g. use of industrial robot manipulators, 3D camera in-hand or hybrid grippers, with preference for suction grasps. While NimbRo developed a robot with simmilar grasping capabilities, their solution was less robust, and partly relied on placing the tote

TABLE II
RESULT SUMMARY OF THE ARC 2016 FINALS.

Team	Total score	Points item+bin	success targets	misplaced items	other penalties
Picking Task					
Team Delft	105	135	9	3 (-30)	0
PFN	105	130	9	1 (-10)	-15
NimbRo Picking	97	152	10	4 (-40)	-15
MIT	67	97	6	2 (-20)	-10
Stowing Task					
Team Delft	214	224	11	1 (-10)	0
NimbRo Picking	186	206	11	2 (-20)	0
MIT	164	164	9	0	0
PFN	161	191	10	3 (-30)	0

below the shelf to drag the products out of the bins. This trick also allowed teams like NimbRo to re-try objects dropping back to the tote during the stow task. However, this implied score penalties as shown in Table II. Actually only Team Delft in Table II did not rely on this "workaround", but instead achieved robust and fast pick&place by attempting only clear picks, moving any occluding objects. In addition, Team Delft's operation maximized the bonus points that were given bsed on the item handled and the filling of the bins, e.g. wiht the selection the stowing bins (see column "Points item+bin" in Table II).

The detailed analysis of performance in Table III shows that the design achieved the system requirements targeted in speed and reliability while addressing the uncertainty conditions presented in section II-B. However, the system presented some limitations that affected its performance specially in the picking task.

1) Detect and locate all the products: The solution for object detection based on deep learning proved highly reliable and fast (avg. 150 ms). It is a *'level 1'* solution that takes additional images from fixed viewpoints on the fly if needed. The solution is robust to varying light conditions, including the dark locations at the back of the bins and the reflections at the front due to products packaging and the metal floor, at the cost of requiring large amounts of training data. On the other hand, it is highly reconfigurable: training the model for a new product requires only a few dozens of images and a few hours. This makes this approach very attractive for tasks where the arrangement of the products is unknown, but sample information can be easily generated.

The pose estimation of the target based on Super 4PCS is less reliable, and its speed had higher variance, due to which a time limit to 4 s was added to trigger fallback mechanisms. Speed could be improved with a GPU implementation of the Super 4PCS algorithm. The main problem for the open-loop solution implemented was the scarce Pointcloud information obtained for certain situations, strongly affected by lighting conditions and packaging reflections. The *'level 1'* fallback mechanism to take additional images from fixed locations was not an improvement. A possible improved *'level 2'* design would use the information from an initial pose estimation to plan a new camera pose. However, in many cases the pose estimation error considered the target in impossible or unreasonable orientations. A more promising enhancement is then to use more application heuristics during design, for example assuming that gravity limits the possible orientations of an object.

2) Stable grasp for all products and orientations: Team Delft's grasping solution was able to pick all 39 items in the 2016 competition, in most orientations and bin locations. The *'level 1'* solution to grasping and product handling, based on a custom gripper, achieved a robust and fast performance for most of the products. The high-flow, low-vacuum combined with a compliant suction cup proved robust to different product surfaces and misalignments ($> 90\%$ success rate). Additionally, it embedded grasp success sensing, providing an interesting standalone *'level 1'* solution. The main problems of the suction mechanism were: inadvertently grasping two objects, and the stability of the grasp for large, heavy products. The first problem could be improved by verifying the grasped products with the tote camera. The second problem is partially addressed by manipulation planning with custom heuristics to orient the product after grasping.

The pinch mechanism is less reliable ($< 50\%$ success rate for the dumbbell). Its lack of compliance demanded a higher accuracy than that provided by the pose estimation module. Additionally, it is an *'level 0'* standalone solution with no feedback on the success of the operation.

3) Reach and manoeuvre all locations: Robot motion is critical in manipulation applications in relation to speed and collisions. Regarding speed, the overall *'level 2'* solution designed allowed to optimize the robot motions during design, achieving a high performance only limited by the grasp stability and safety[4]. As an indicator, Team Delft robot achieved an average cycle time of 35 s, compared to more than a minute for the feedback-based winning solution in 2015 [11].

In relation to reliability, Team Delft solution combined offline information in the 3D models of the workspace and the runtime Pointcloud information from the 3D cameras to avoid collisions. Avoiding collisions guarantees not modifying the structure of the environment, thus facilitating the application of an open-loop *'level 2'* solution. However, it is more limited for handling cluttered situations, where it can be unavoidable to touch objects next to the target. Additionally, it is more sensitive to perception errors and scarcity of data, as is the case of the shelf, where a limited range of viewpoints is possible.

4) Overall Solution: The overall *'level 2'* sense-plan-act solution achieves a high-performance when the core assumption of an accessible grasp surface holds. This is the case in the stowing task, a standard bin picking application, thanks to gravity and task conditions (workspace geometry and product characteristics).

Bin cluttering presented the main difficulty to the *'level 2'* approach, making the collision-free requirement hard to address. Avoiding it by moving items around posed the disadvantage that an unknown number of pick and place actions could be needed to remove all the objects occluding the grasp of the target item. On the other hand, the simplification to a

[4]Due to the competition setup the robot speed limits were set to a safe 0.5 factor of its nominal maximum joint speed.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2018.2800744, IEEE Transactions on Industrial Informatics

TABLE III
SYSTEM PERFORMANCE SUMMARY

44 pick&place operations attempted in the Picking and Stowing finals. 38 operations on target items, and 6 to move occluding items.			
Successful		25	11.35 s (avg) sense&plan 22.41 s (avg) act (robot motions) 33.76 s (avg) total pick&place execution time
Failed	recovered	3	target not detected or pose not estimated
		10	no motion plan found
		6	target not held after grasp
	penalties	1	target dropped outside the bin
		1	non-target shoved outside the bin

System fatal errors	Stall condition due to no more feasible targets. Emergency stop due to gripper crushing object.

'level 2' system using only two primitive actions allowed us to optimize the required robot motions for speed, by exploiting the structure of task and environment as described in section IV-E.

Another disadvantage of our *'level 2'* approach was the limited compliance with runtime uncertainty and its need for accuracy. The localization of the target products has to be precise to 5 mm, as well as the collision information. The hardware selection regarding the cameras and the gripper design proved that it is critical to simplify the control solution.

In relation to workspace uncertainty, the solution proved robust to deviations in the shelf's geometry, they being due to construction or to the 3 cm displacements allowed by the competition rules. These uncertainties were compensated for by the shelf pose estimation procedure performed during the sense cycle.

5) Failure Management: The possible failures of the system are: i) the product cannot be picked ii) product is dropped, iii) critical collision leading to equipment or product damage. These failures could arise from any of the core modules of the Team Delft-APC system namely Vision, Grasping or Motion. While exhaustive testing of all failure modes on the final system was difficult to perform due to practical reasons, some of the failures observed while testing and the actual run in the final competition are listed in Table III.

The nature of the failures caused by the core module Vision were fairly consistent in the products and the situation that caused them. However, the failures caused by the core modules Grasping and Motion were difficult to reproduce as they were caused by inverse kinematic solutions that would put the last few joints of the robot in a singular configuration while performing specific grasps (such as Bubble mailer leaning on the side walls of the top left or right corner bins). This was mainly caused by the numerical optimization based TRAC-IK solver used for the grasp motion planning. This non-determinism could have perhaps been reduced by using a fixed seed for the TRAC-IK solver, but, we did not have the time to implement and test this solution before the challenge.

The error handling mechanisms described in IV-F provided for reliability when the product cannot be picked, by either retrying it under different conditions (new camera images), or postponing that target. When the product is dropped, appropriate action or reporting was coded using favorable assumptions about the result of the action.

'Level 1' mechanisms for error handling specific to the application are unavoidable. However, general methods and tools that allow to capture them in a scalable and maintainable way are critical, such as the state-machine tool SMACH used by Team Delft.

B. Lessons Learned

Overall Team Delft's approach to design the level of automation required for each problem proved to be successful, outperforming concurring and previous editions' entries in the Amazon Robotics Challenge. The main lessons learned relate to the value of open-loop solutions, how deep learning can contribute to them, and the need for collisions in manipulation.

In semi-structured, closed environments, the proper combination of open-loop solutions and hardware tailoring based on adequate assumptions provides the best performance. An exemplary case that proves that exploring simplifying hypothesis with open-loop solutions is worthy are the deformable products in the Amazon competition. Initially, they seemed to pose problems to locate, given the lack of a 3D model for them, and even more difficulties to manipulation planning. However, detection worked flawlessly with enough training data, and the compliant and powerful suction gripper made precise localization and careful manipulation unnecessary.

To address the limited flexibility and corner cases, Team Delft integrated two different solutions that exploit application knowledge at design time to tackle runtime uncertainty. For grasping, manipulation, and error handling, heuristics were programmed. They provide a simple and powerful solution easy to implement and inspect. However, their practical implementation present important problems. First, they require intensive work by experts. They are hardly reusable. Finally, they do not scale: in complex applications such as the Amazon Robotics Challenge, where there are many robot technologies and components connected, modifying or adding new task-level heuristics in the design becomes unmanageable. The challenge remains to find practical ways to systematically encode knowledge, something the AI and cognitive robotics communities have been targeting for decades.

Deep learning techniques are a very promising solution to automate the generation of application-specific solutions embedding task-knowledge. Despite being an open-loop solution, automated training allows to easily accommodate for variations (bounded uncertainties), as well as to easily adapt it to new products or environments. In Team Delft's design, deep learning proved a very reliable and performing solution for object detection. Another entry in 2016 competition[5], as well as more recent results [31], [32] suggest that deep learning techniques can be successfully applied to grasping. However, the generation of training data in this case is very resource consuming. An intermediate, more feasible solution could be applying it to pose estimation.

[5]https://www.preferred-networks.jp/en/news/amazon-picking-challenge_result

Grasping from the bin poses the more difficulties for Team Delft's open-loop solution. Main cause is the rejection of grasp plans due to collisions. Many results suggest that grasping and manipulation require to allow for contact, using techniques that incorporate force/torque information [11]. Feedback solutions seem unavoidable for successful manipulation in cluttered environments. However, for improved performance it is desirable to limit their scope in the robot control by combining them with the better performing planning solutions. Current robot motion planning based on joint configuration space presents problems with grasp stability and does not allow for more flexible online planning based on force/torque feedback. Kinodynamic motion planning can overcome these limitations, but more research is needed for it to become a practical and feasible solution.

VI. Conclusion

This paper provides a practical discussion on the challenges for industrial robot manipulation for product handling, based on the experience of the authors developing the winning solution in the Amazon Robotics Challenge 2016. From this experience: 1) the specific task conditions should guide the selection of the robotic solutions for an application, 2) understanding the characteristics of the solutions chosen and their relation to the task's conditions, embedded in multiple design decisions and assumptions, is critical for the performance of the overall system integrated from them, and 3) this characterization can be done according to 'robot automation levels', based on the use of information to address the task uncertainties during the development and runtime of the robotic system. The previous considerations guided the development of Team Delft's robotic system, which achieved an mean pick&place time of 33.76 s, correctly handling 43 out of 44 targets in a semi-structured environment.

References

[1] "Amazon robotics challenge," May 2017. [Online]. Available: https://www.amazonrobotics.com/

[2] N. Correll, K. E. Bekris, D. Berenson, O. Brock, A. Causo, K. Hauser, K. Okada, A. Rodriguez, J. M. Romano, and P. R. Wurman, "Analysis and Observations From the First Amazon Picking Challenge," *IEEE Transactions on Automation Science and Engineering*, vol. PP, no. 99, pp. 1–17, 2017.

[3] E. Ackerman, "German Warehouse Robots Tackle Picking Tasks," https://spectrum.ieee.org/automaton/robotics/industrial-robots/german-warehouse-robots-tackle-picking-tasks, 2016.

[4] I. Robotiq, "Robotiq, Inc." [Online]. Available: https://robotiq.com

[5] Empire Robotics, Inc. VERSABALL. [Online]. Available: http://empirerobotics.com/

[6] L. U. Odhner, L. P. Jentoft, M. R. Claffee, N. Corson, Y. Tenzer, R. R. Ma, M. Buehler, R. Kohout, R. D. Howe, and A. M. Dollar, "A compliant, underactuated hand for robust manipulation," *The International Journal of Robotics Research*, vol. 33, no. 5, pp. 736–752, 2014.

[7] R. Deimel and O. Brock, "A novel type of compliant and underactuated robotic hand for dexterous grasping," *The International Journal of Robotics Research*, vol. 35, no. 1-3, pp. 161–185, 2016.

[8] Y. Hua, S. Zander, M. Bordignon, and B. Hein, "From AutomationML to ROS: A model-driven approach for software engineering of industrial robotics using ontological reasoning," in *2016 IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA)*, Sept 2016, pp. 1–8.

[9] M. Moon, "Amazon crowns winner of first warehouse robot challenge," Engadget, 2015.

[10] M. Quigley, K. Conley, B. P. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "ROS: an open-source Robot Operating System," in *ICRA Workshop on Open Source Software*, 2009.

[11] C. Eppner, S. Höfer, R. Jonschkowski, R. Martín-Martín, A. Sieverling, V. Wall, and O. Brock, "Lessons from the Amazon Picking Challenge: Four Aspects of Building Robotic Systems," in *Robotics: Science and Systems XII*, 2016.

[12] A. Robotics, "Amazon Robotics Challenge 2016." [Online]. Available: https://www.amazonrobotics.com/#/roboticschallenge/past-challenges

[13] A. ten Pas and R. Platt, "Using Geometry to Detect Grasp Poses in 3D Point Clouds," in *Proceedings of the International Symposium on Robotics Research (ISRR)*, 2015.

[14] A. W. D. Fischinger and M. Vincze, "Learning grasps with topographic features," *The International Journal of Robotics Research*, vol. 34, no. 9, pp. 1167–1194, May 2015.

[15] J. M. Q. Lei and M. Wisse, "A survey of unknown object grasping and our fast grasping algorithm c-shape grasping," in *Proceedings of the IEEE International Conference on Control, Automation and Robotics (ICCAR)*, 2017, pp. 150–157.

[16] R. Parasuraman, T. B. Sheridan, and C. D. Wickens, "A model for types and levels of human interaction with automation," *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 30, no. 3, pp. 286–297, May 2000.

[17] R. Brooks, "Intelligence without representation," *Artificial intelligence*, vol. 47, no. 1-3, pp. 139–159, 1991.

[18] E. F. Camacho and C. B. Alba, *Model predictive control*. Springer Science & Business Media, 2013.

[19] J. Dai, Y. Li, K. He, and J. Sun, "R-FCN: Object Detection via Region-based Fully Convolutional Networks," in *Advances in Neural Information Processing Systems 29*, D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, Eds. Curran Associates, Inc., 2016, pp. 379–387. [Online]

[20] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, June 2017.

[21] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587.

[22] J. R. Uijlings, K. E. Van De Sande, T. Gevers, and A. W. Smeulders, "Selective search for object recognition," *International journal of computer vision*, vol. 104, no. 2, pp. 154–171, 2013.

[23] R. Girshick, "Fast R-CNN," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1440–1448.

[24] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.

[25] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[26] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks." in *Aistats*, vol. 9, 2010, pp. 249–256.

[27] N. Mellado, D. Aiger, and N. J. Mitra, "Super 4PCS Fast Global Pointcloud Registration via Smart Indexing," *Computer Graphics Forum*, vol. 33, no. 5, pp. 205–215, 2014.

[28] P. J. Besl and N. D. McKay, "A method for registration of 3-D shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 239–256, Feb 1992.

[29] C. Hernández, M. Bharatheesha, W. Ko, H. Gaiser, J. Tan, K. van Deurzen, M. de Vries, B. V. Mil, J. van Egmond, R. Burger, M. Morariu, J. Ju, X. Gerrmann, R. Ensing, J. van Frankenhuyzen, and M. Wisse, Team Delft's Robot Winner of the Amazon Picking Challenge 2016, in S. Behnke, R. Sheh, S. Sarıel, and D. D. Lee, eds., *RoboCup 2016 Proceedings (to appear)*, volume 9776 of *Lecture Notes in Computer Science*, pp. 613–624, Springer, 2017.

[30] I. A. Sucan and S. Chitta. (2017, September) "MoveIt!". [Online]. Available: http://moveit.ros.org

[31] S. Levine, P. Pastor, A. Krizhevsky, and D. Quillen, "Learning Hand-Eye Coordination for Robotic Grasping with Deep Learning and Large-Scale Data Collection," *CoRR*, vol. abs/1603.02199, 2016.

[32] J. Mahler and K. Goldberg, "Learning Deep Policies for Robot Bin Picking by Simulating Robust Grasping Sequences," in *Proceedings of the 1st Annual Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, S. Levine, V. Vanhoucke, and K. Goldberg, Eds., vol. 78. PMLR, 13–15 Nov 2017, pp. 515–524.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2018.2800744, IEEE Transactions on Industrial Informatics

Carlos Hernández Corbato is a postdoctoral researcher at the TU Delft Robotics Institute, Delft University of Technology. Carlos is currently coordinator of the ROSIN European project granted in the H2020 program. He has participated in other national and European projects in the topics of cognitive robotics and factories of the future. Carlos graduated with honors in Industrial Engineering (2006) and received his M.Sc. Ph.D in Automation and Robotics from the Universidad Politcnica de Madrid in 2013. Carlos' research interests include: cognitive architectures, autonomy and model-based engineering.

Mukunda Bharatheesha is a Ph.D. candidate at the TU Delft Robotics Institute. He obtained his M.Sc. degree in Embedded Systems with a specialization in Control Systems from Delft University of Technology in 2011. He subsequently worked as a researcher at the Delft Biorobotics Lab for a period of 1 year where he focused on path planning and autonomous navigation for mobile robots. His current research focuses on motion planning with dynamical constraints for serial link manipulators. He is exploring supervised learning as a tool for speeding up the process of generating dynamically feasible motions using sampling-based motion planning algorithms.

Jeff van Egmond is a Ph.D. candidate at the TU Delft Robotics Institute, Delft University of technology. Jeff is involved in the Factory-in-a-Day European project. Jeff graduated in Mechanical Engineering (2014) at Delft University of Technology. Jeff's research interests are focused on applying camera systems in automation and robotics. Jeff participated in Team Delft, winning the Amazon Robotics Challenge 2016. He worked on early development (2012) of the IHMC's second place finish at the DARPA Robotics Challenge in 2015.

Jihong Ju received the B.S. degree in Physics from Shandong University, Jinan, China, and the M.Sc. degree in Physics from the University of Manchester, Manchester, UK, in 2014 and 2015, respectively. He worked as an intern at Delft Robotics in 2016 and, in the same year, started as an intern at IBM Center for Advanced Studies Benelux, Amsterdam, Netherlands.

Martijn Wisse received the M.Sc. and Ph.D. degrees in mechanical engineering from the Delft University of Technology, Delft, The Netherlands, in 2000 and 2004, respectively. He is currently a Professor with the Delft University of Technology. His previous research focused on passive dynamic walking robots, and now he is using concepts from that research (such as passive stability) for the field of robot manipulators for agile manufacturing. His current research interests include underactuated grasping, open-loop stable manipulator control, design of robot arms and robotic systems, agile manufacturing, and the creation of startup companies in the same field.