Lessons from the Amazon Picking Challenge: Four Aspects of Building Robotic Systems

Clemens Eppner[*], **Sebastian Höfer**[*], **Rico Jonschkowski**[*], **Roberto Martín-Martín**[*], **Arne Sieverling**[*], **Vincent Wall**[*], and **Oliver Brock**
Robotics and Biology Laboratory, Technische Universität Berlin, Germany
{clemens.eppner,sebastian.hoefer,rico.jonschkowski,roberto.martinmartin,
arne.sieverling,vincent.wall,oliver.brock}@tu-berlin.de

Abstract

We describe the winning entry to the Amazon Picking Challenge 2015. From the experience of building this system and competing, we derive several conclusions: *1)* We suggest to characterize robotic system building along four key aspects, each of them spanning a spectrum of solutions—modularity vs. integration, generality vs. assumptions, computation vs. embodiment, and planning vs. feedback. *2)* To understand which region of each spectrum most adequately addresses which robotic problem, we must explore the full spectrum of possible approaches. *3)* For manipulation problems in unstructured environments, certain regions of each spectrum match the problem most adequately, and should be exploited further. This is supported by the fact that our solution deviated from the majority of the other challenge entries along each of the spectra. *This is an abridged version of [Eppner* et al.*, 2016].*

1 Introduction

The *Amazon Picking Challenge 2015* (APC) tested the ability of robotic systems to fulfill a fictitious order by autonomously picking the ordered items from a warehouse shelf (Fig. 1). The system presented here outperformed the 25 other entries, winning by a significant margin. In this paper, we provide a technical description and experimental evaluation of our system. Our system-building experience led to the following insights: Robotic systems can be characterized along *four key aspects*. To develop a shared understanding of system building we should *explore these spectra*. For *manipulation in unstructured environments*, we believe that certain regions of each spectrum match the problem characteristics most adequately and should be examined by roboticists with increased emphasis. Those four key aspects are:

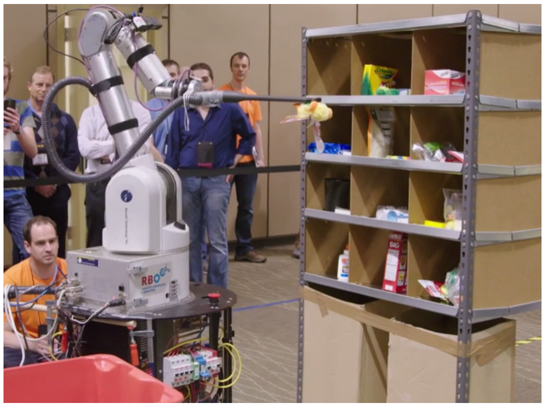

Figure 1: Our robot picks a plush toy during the challenge.

A. Modularity vs. Integration: In robotics, success is determined by the behavior of the entire system, not by individual modules [Brooks, 1990]. Still, a high degree of modularity allows breaking down problems into simpler subproblems. Wrong modularization, however, can make solving problems unnecessarily difficult. Until we fully understand which modularization is most adequate for manipulation in unstructured environments, we suggest to build tightly integrated systems and constantly revise their modularization.

B. Computation vs. Embodiment: Robot behavior results from the interplay of computation (software) and embodiment (hardware). Computation is a powerful and versatile tool but adapting the embodiment sometimes leads to simple and robust solutions. We suggest that in manipulation, one should consider alternative embodiments as part of the solution process.

C. Planning vs. Feedback: Planning performs search in a world model, leading to verifiable solutions. Feedback from physical interactions, on the other hand, reduces uncertainty and allows to find local solutions without expensive computation. We suggest to use planning only when necessary and explore the use of feedback as an alternative when the manipulation task does not require global search.

D. Generality vs. Assumptions: For robotics research, finding general solutions is highly desirable. However, solving

[*]Authors contributed equally to this work and are ordered alphabetically. We gratefully acknowledge the funding provided by the Alexander von Humboldt foundation and the Federal Ministry of Education and Research (BMBF), the European Commission (SOMA project, H2020-ICT-645599), the German Research Foundation (DFG) (Exploration Challenge, BR 2248/3-1), and the travel grant provided by Amazon Robotics. We would like to thank Barrett Technology for their support and our team members Raphael Deimel, Roman Kolbert, Gabriel Le Roux, and Wolf Schaarschmidt, who helped creating the winning system.

the most general problem might be unnecessary or even unfeasible. We suggest to search for reasonable and useful assumptions that aid solving manipulation problems in unstructured environments. By extracting, sharing, and revising assumptions that prove useful for an increasingly broader variation of a problem, we will naturally progress towards a general solution.

These aspects are not novel and certainly will not surprise the robotic practitioner. However, what should come as a surprise is the sparsity with which the corresponding spectra have been explored by our community and how rarely these aspects are used explicitly to characterize robotic systems. Case in point: Our solution to the APC explores different regions on these spectra than most other challenge entries [Correll *et al.*, 2016]. We believe that these differences were crucial for our success.

We propose that by making the four key aspects of robotic systems (and possibly additional ones that we did not identify yet) explicit, our community will begin to understand the mapping of problem characteristics to the appropriate regions on these spectra. Our paper is, of course, only a single data point in this endeavor. But if our community starts characterizing robotic systems along the proposed axes, thereby making design choices transparent and comparable, we might move towards a scientific theory of system building [Atkeson *et al.*, 2015; Hawes *et al.*, 2010; Katz and Brock, 2011].

2 The Amazon Picking Challenge 2015

The APC consists of autonomously picking twelve out of 25 objects from a warehouse shelf and placing them into a storage container (Fig. 1) within 20 minutes. The robot knows which objects are contained in each of the shelf's twelve bins, but not of their exact arrangement. For each successfully picked target object, the robot receives 10, 15, or 20 points, depending on how many additional objects were in the same bin. The 25 objects varied widely in size and appearance.

3 Technical System Description

We now describe the hardware and algorithmic components of our solution. We will mention connections between our design choices and the four key aspects.

Hardware Components We use a 7-DoF Barrett WAM mounted on a Nomadic XR4000 mobile base. The inclusion of holonomic mobility—a choice of embodiment that set our solution apart from most other entries in the APC—greatly facilitated the generation of motion. The ability to reposition the base enabled the arm to easily reach inside all of the bins. Our end-effector consists of a modified crevice nozzle with a suction cup mounted at its tip. An off-the-shelf vacuum cleaner generates sufficient air flow to lift up to $1.5\,kg$. It can reliably pick up all challenge objects except for the pencil cup. Grasping success is rather insensitive to the exact contact location with the object, leading to reduced requirements for perception. At the same time, the end-effector's thin shape reduces the need for complex collision avoidance, as it easily fits in between objects, pushing them aside if necessary. This simple choice for the end-effector illustrates that an appropriate embodiment simplifies different aspects of the overall solution, including grasp planning and perception.

Motion Generation Objects are picked using two predefined grasp strategies: a top-down grasp, and a grasp approaching the object from the side. Both primitives deliberately move the end-effector *into* the object and push it against the floor, walls, or other objects. This is an example of exploiting the environment to guide manipulation [Eppner *et al.*, 2015] using haptic feedback. The execution of picking motions is realized with continuous feedback controllers. We transition between these controllers based on discrete sensor events. This behavior can be described by hybrid automata [Egerstedt, 2000]. The automaton also contains countermeasures for common failures. The resulting hybrid automaton consists of 26 states and 50 transitions, of which 34 deal with error handling, e.g. if the robot detects an undesired contact with the shelf, it retracts from the bin and reattempts to pick the object later.

Object Recognition Based on data from an RGB-D camera three steps are performed: feature extraction, object segmentation, and bounding box fitting. The first step extracts a number of task-specific features for every pixel of the RGB-D image. These features include information about color, distance to the tracked shelf model and height within the bin. Instead of searching for features that could solve the general object recognition problem, these task-specific features rely on strong assumptions (e.g. that objects are placed in a known shelf). Statistics about these pixel-features for each object are derived from manually segmented training images and enable the second step to find the image segment that has the highest probability of belonging to the target object. This step exploits additional assumptions to facilitate segmentation, e.g. that only a small subset of all objects is present in every bin and that the robot knows which objects these are. The third step takes the point cloud for this segment and fits a bounding box. This allows the robot to decide where and from which direction it should perform the pick. Our object recognition pipeline is described in [Jonschkowski *et al.*, 2016].

4 Evaluation

The APC provides an in-depth evaluation of our system, comparing its performance with 25 teams from around the world. We complement these results with nine additional experiments, using object configurations from the competition.

Quantitative Evaluation During the competition we scored 148 out of 190 points (2^{nd}: 88, 3^{rd}: 35). We attempted to pick all twelve objects and were successful for ten. An average picking motion took 87 seconds. This allowed us to maximally attempt 14 picks within the 20 minutes of challenge duration.

We reenacted all five shelf configurations that were used in the challenge for further testing. We performed two trials per setup, using the robot system from the challenge without modifications. During 200 minutes the robot picked 95 objects, of which 85 were target objects. On average we collected 117.6 points ($\sigma = 29.2$) which is 62.5% of all available

85	successful picks
13	object recognition failures
9	bulky objects stuck at removal
9	small objects (end effector imprecision)
2	displacing objects during approach
2	meshed pencil cup (suction fails)

Table 1: Failure cases for 120 picking attempts

ones. This shows that the competition run was on the upper end of the system's capabilities. Still only one (72 points) out of ten trials would have lead to our team placing second.

System Limitations From the 120 attempted picks, the system picked ten wrong objects and failed to pick 25 objects. We think that all failure cases (Table 1) can be addressed by shifting along the spectra of the proposed aspects.

Object Recognition Failures: We attribute 13 failed picking attempts to the object recognition pipeline. These failures occur when our local features cannot discriminate between the objects present in the target bin, resulting in wrong object boundaries or mistaking another object for the target object.We believe that object recognition can be improved most effectively by shifting along spectrum A towards tighter integration, and along spectrum C towards more feedback. We could reject poses that result in physically implausible configurations by tighter integration of segmentation and geometric pose reasoning. Moreover, we could decrease the likelihood of picking wrong objects by weighing objects or visually inspecting them after the pick.

Bulky Objects Stuck at Removal: Eight scenarios contained a large box which could only be removed from the bin by tilting it. Our system failed on all attempts. The long bottle brush also got stuck once on the shelf lip and dropped. To address these failures, we need to shift along spectrum C towards (motion) planning. Planning would allow us to reason how to reorient objects to remove them from the bin.

Small Objects: Out of ten attempts, the robot failed nine times attempting to pick up the small spark plugs. In the competition run, the robot even picked up a non-target object instead. These failures result from the fact that the reaching movement is executed open-loop, accumulating a significant error. This can be addressed by shifting along spectrum C towards more feedback, e.g. by using visual servoing.

Displacing Objects: In five out of ten attempts, the robot toppled over the glue bottle. The bottle then required a reattempt from the top. In two cases the robot did not have enough time for a reattempt and lost points. As before, this failure case can be alleviated by additional feedback; tumbling could be detected earlier and lead to a different strategy.

Pencil Cup: The meshed metal pencil cup does not have enough solid surface to pick it with suction. This failure mode shows a limitation of our chosen embodiment (Sec. 5.2). It suggests possible extensions to our end-effector, e.g. adding a mechanical or magnetic gripper.

5 Key Aspects of Building Robotic Systems

We will now generalize our experience from the APC to building robotic systems in general. For each aspect we present arguments and examples for both ends of the spectrum and position our system on it.

5.1 Modularity vs. Integration

There is a continuum between tightly integrated and modular solutions which has been investigated in systems engineering, computer science and product management.

Modularity Building systems of arbitrary complexity without structuring them into modules is difficult. Modularity decomposes complexity by breaking down a problem into smaller sub-problems that can be solved and tested individually. Because of the power of compositionality, building modular systems is the prevalent paradigm in robotics. This is reflected in the separation of robotics into the classical fields of perception, planning, control, etc. as well as in the produced software. For example, high modularity is one of the core concepts of ROS [Quigley *et al.*, 2009], a popular framework for implementing robotic systems. Similarly, libraries like OpenCV, PCL, and MoveIt! represent commonly employed modules for computer vision and planning.

Integration Robotic systems generate behavior as a result of integrating many software and hardware components [Brooks, 1990; Cohen, 1996]. The usefulness of a robotic system is determined by the performance of the integrated system, rather than by the performance of individual components. To ensure that the performance of the entire system is maximized, and to avoid making wrong commitments or addressing sub-problems that are unnecessarily difficult, all components of the system should be chosen to maximally exploit potential synergies between components. To identify these synergies in the absence of established system-building guidelines requires early integration [Johnson *et al.*, 2015; Katz and Brock, 2011]. Important advances were achieved by overcoming existing "modularizations", e.g. by combining interaction and perception [Bohg *et al.*, 2016; Martín-Martín and Brock, 2014].

Our Design Choice on the Spectrum Our system used ROS [Quigley *et al.*, 2009] and relied on various standard modules, e.g. for visual processing and navigation. However, we embraced tight integration at various levels. We integrated planning and control using hybrid automata, adapted our picking strategies to the embodiment and the requirements for object recognition to the picking strategies. Furthermore, we adapted many ideas from agile development [Schwaber, 2004]: rapid prototyping, early and continuous integration, adversarial testing, and shared knowledge.

5.2 Computation vs. Embodiment

The idea that mechanisms and materials expose behavior that is normally attributed to computation is known as *morphological computation* [Pfeifer and Gómez, 2009].

Computation Since computation is more flexible and can be altered easily, compared to the embodiment (hardware), it allows building highly complex systems with diverse behaviors. Purely computational approaches to robotics [Coleman, 2015; LaValle, 2006] also have the advantage of potentially being hardware-agnostic. Many examples of computation-focused approaches exist in robotics, e.g. posing grasping as a contact point planning problem [Miller and Allen, 2004]. It is appealing as it abstracts the problem away from the hand and the environmental context. Similarly, perception has been traditionally seen as a passive, purely computational problem [Marr, 1982].

Embodiment Tailoring the hardware to a particular problem can reduce the required computation. Hardware solutions are often simple and robust, especially when uncertainty is present. Grasp planning can benefit substantially from embodiment, as exemplified by simple under-actuated robotic hands [Deimel and Brock, 2016; Dollar and Howe, 2010]. Appropriate embodiment also facilitates perception, e.g. adaptive hands reduce the need for accuracy in object pose estimation. Moreover, placing a vision sensor on the robot arm increases the sensor's field of view and reduces the effect of occlusions [Aloimonos *et al.*, 1988].

Our Design Choice on the Spectrum We reduced the need for computation by using a suction cup. The *reduced* number of degrees of freedom simplified grasp planning and object pose estimation. We also reduced the need for computation by *increasing* the number of degrees of freedom by mounting the robot arm on a mobile base. This allowed us to generate motion mostly through feedback control, rather than resorting to motion planning. However, in the APC we failed to pick the pencil cup due to our chosen embodiment.

5.3 Planning vs. Feedback

Classical robotics and AI employed the sense-plan-act paradigm, assuming the robot can build a perfect model of the world. The difficulty of obtaining such models initiated a shift towards feedback-driven approaches [Brooks, 1990].

Planning Planning finds global solutions, where controllers based on local feedback would fail. The most common application in robotics is motion planning [LaValle, 2006]. Practitioners need to provide models of the environment, calibrate the robot, and localize it in the environment [Thrun *et al.*, 2005]. Under these prerequisites, motion planners serve as general and versatile black-box solvers.

Feedback If global search is not required or not possible, feedback control based on task-relevant features is often sufficient to generate successful robot motion. Feedback can be exploited in the visual [Espiau *et al.*, 1992] or contact domain [Lozano-Perez *et al.*, 1984; Eppner *et al.*, 2015] to simplify manipulation tasks. Feedback approaches are particularly useful in the presence of uncertainty, high dimensionality, long time horizons, and inaccurate models. In these cases, planning would be computationally demanding and often intractable [Papadimitriou and Tsitsiklis, 1987].

Our Design Choice on the Spectrum Our system relies on very simple planning. We use on-line grasp approach planning and execute the motions using pre-defined, feedback-guided motion primitives, avoiding configuration-space motion planning altogether. This positions our solution far to the feedback-side of the spectrum, in contrast to the majority of other challenge entries (80% of the teams used motion planning, 44% used MoveIt! [Sucan and Chitta, 2016], [Correll *et al.*, 2016]). Feedback control is so successful in the APC setting because the task only requires a limited range of motions and the shelf provides plenty of contact surfaces. However, some shortcomings of our system, such as the lack of in-bin reorientation of objects, should be addressed by some form of planning.

5.4 Generality vs. Assumptions

This spectrum is reflected in the no free lunch theorem [Wolpert, 1996], the bias-variance trade-off [Hastie *et al.*, 2005], and the balance between generality and specificity in system design [van Gigch, 1991].

Generality Finding general solutions is an important goal in robotics. A general solution applies to a wide range of problems and reflects a deep understanding of the problem at hand. In contrast, solutions strongly tailored to a specific problem instance (e.g. a robot demo) might not lead to insights or contribute to a broader understanding of system building. A number of general approaches were successfully applied in robotics, e.g. task-generic planning algorithms such as A^* are widely used for mobile robot navigation. Recursive Bayesian estimation is a very generic framework that helped solving many different problems in robotics.

Assumptions In machine learning, search, and optimization, the no free lunch theorems prove that no problem can be solved without making appropriate assumptions [Wolpert, 1996; Wolpert and Macready, 1997]: averaged over all possible problems, there is no method that outperforms random guessing. The only way to improve on random guessing is by making assumptions about the problem. We believe that problems in robotics are characterized by a significant amount of reoccurring underlying structure. E.g. in motion planning, adding information about workspace connectivity can reduce the computational complexity by up to three orders of magnitude [Rickert *et al.*, 2014]. In reinforcement learning, adding explicit knowledge about physics makes the learning problem tractable by reducing its dimensionality [Jonschkowski and Brock, 2015; Scholz *et al.*, 2014].

Our Design Choice on the Spectrum Our APC system used available (general) solutions whenever they proved sufficient to solve the problem. Since we could not find existing approaches for reliably locating the target object, we used various assumptions to simplify the problem, e.g. that the objects are placed in a known shelf or that the contents of each shelf bin are known and therefore only a small number of objects need to be considered. Our general solutions included a particle filter for localizing the robot, and standard joint and operational space controllers for motion generation.

6 Conclusion

We presented and evaluated our winning system for the 2015 APC. To describe the system, we proposed four key aspects of system building. A systematic description of robotic systems according to these aspects (and additional ones proposed by others in the future) will facilitate accumulating general knowledge for building robotic systems.

Our lessons on building robotic systems are consistent with those others derived from their experience in similar robotics challenges. In the area of autonomous driving [Buehler *et al.*, 2007] and humanoid robotics [Atkeson *et al.*, 2015], the resulting insights have led to significant advances. We hope that the APC and our lessons learned will be equally useful for manipulation.

References

[Aloimonos *et al.*, 1988] J. Aloimonos, I. Weiss, and A. Bandyopadhyay. Active vision. *International Journal of Computer Vision*, 1(4):333–356, 1988.

[Atkeson *et al.*, 2015] C. G. Atkeson, B. P. W. Babu, N. Banerjee, D. Berenson, C. P. Bove, X. Cui, M. DeDonato, R. Du, S. Feng, P. Franklin, et al. No falls, no resets: Reliable humanoid behavior in the DARPA robotics challenge. In *Humanoids*, pages 623–630, 2015.

[Bohg *et al.*, 2016] J. Bohg, K. Hausman, B. Sankaran, O. Brock, D. Kragic, S. Schaal, and G. Sukhatme. Interactive perception: Leveraging action in perception and perception in action. *IEEE Transactions on Robotics*, 2016. Under review.

[Brooks, 1990] R. A. Brooks. Elephants don't play chess. *Robotics and Autonomous Systems*, 6(1):3–15, 1990.

[Buehler *et al.*, 2007] M. Buehler, K. Iagnemma, and S. Singh. *The 2005 DARPA grand challenge: the great robot race*, volume 36. Springer Science & Business Media, 2007.

[Cohen, 1996] P. R. Cohen. Empirical methods for artificial intelligence. *IEEE Intelligent Systems*, 11(6):88, 1996.

[Coleman, 2015] D. Coleman. MoveIt! strengths, weaknesses, and developer insight. Presentation at ROSCon, 2015.

[Correll *et al.*, 2016] N. Correll, K. E. Bekris, D. Berenson, O. Brock, A. Causo, K. Hauser, K. Okada, A. Rodriguez, J. M. Romano, and P. R. Wurman. Lessons from the Amazon Picking Challenge. *ArXiv e-prints*, January 2016.

[Deimel and Brock, 2016] R. Deimel and O. Brock. A novel type of compliant and underactuated robotic hand for dexterous grasping. *The International Journal of Robotics Research*, 35:161–185, 2016.

[Dollar and Howe, 2010] A. M. Dollar and R. D. Howe. The highly adaptive SDM hand: Design and performance evaluation. *The International Journal of Robotics Research*, 29(5):585–597, 2010.

[Egerstedt, 2000] M. Egerstedt. Behavior based robotics using hybrid automata. In *Hybrid Systems: computation and control*, pages 103–116. Springer, 2000.

[Eppner *et al.*, 2015] C. Eppner, R. Deimel, J. Álvarez Ruiz, M. Maertens, and O. Brock. Exploitation of environmental constraints in human and robotic grasping. *The International Journal of Robotics Research*, 34(7):1021–1038, 2015.

[Eppner *et al.*, 2016] C. Eppner, S. Höfer, R. Jonschkowski, R. Martın-Martın, A. Sieverling, V. Wall, and O. Brock. Lessons from the amazon picking challenge: Four aspects of building robotic systems. *Proc. of RSS*, 2016.

[Espiau *et al.*, 1992] B. Espiau, F. Chaumette, and P. Rives. A new approach to visual servoing in robotics. *IEEE Transactions on Robotics and Automation*, 8(3):313–326, 1992.

[Hastie *et al.*, 2005] T. Hastie, R. Tibshirani, J. Friedman, and J. Franklin. The elements of statistical learning: data mining, inference and prediction. *The Mathematical Intelligencer*, 27(2):83–85, 2005.

[Hawes *et al.*, 2010] N. Hawes, M. Zillich, and P. Jensfelt. Lessons learnt from scenario-based integration. In *Cognitive Systems*, pages 423–438. Springer, 2010.

[Johnson *et al.*, 2015] M. Johnson, B. Shrewsbury, S. Bertrand, T. Wu, D. Duran, M. Floyd, P. Abeles, D. Stephen, N. Mertins, A. Lesman, et al. Team IHMC's lessons learned from the DARPA robotics challenge trials. *Journal of Field Robotics*, 32(2):192–208, 2015.

[Jonschkowski and Brock, 2015] R. Jonschkowski and O. Brock. Learning state representations with robotic priors. *Autonomous Robots*, 39(3):407–428, 2015.

[Jonschkowski *et al.*, 2016] R. Jonschkowski, C. Eppner, S. Höfer, R. Martín-Martín, and O. Brock. Probabilistic multi-class segmentation for the Amazon Picking Challenge. In *IEEE/RSJ ICRA*, 2016.

[Katz and Brock, 2011] D. Katz and O. Brock. A factorization approach to manipulation in unstructured environments. In *ISRR 2009*, pages 285–300. Springer, 2011.

[LaValle, 2006] S. M. LaValle. *Planning Algorithms*. Cambridge University Press, New York, NY, USA, 2006.

[Lozano-Perez *et al.*, 1984] T. Lozano-Perez, M. T. Mason, and R. H. Taylor. Automatic synthesis of fine-motion strategies for robots. *The International Journal of Robotics Research*, 3(1):3–24, 1984.

[Marr, 1982] D. Marr. *Vision: A computational approach*. Freeman & Co., San Francisco, 1982.

[Martín-Martín and Brock, 2014] R. Martín-Martín and O. Brock. Online interactive perception of articulated objects with multi-level recursive estimation based on task-specific priors. In *IROS*, pages 2494–2501, 2014.

[Miller and Allen, 2004] A. T. Miller and P. K. Allen. GraspIt! a versatile simulator for robotic grasping. *IEEE Robotics & Automation Magazine*, 11(4):110–122, 2004.

[Papadimitriou and Tsitsiklis, 1987] C. H. Papadimitriou and J. N. Tsitsiklis. The complexity of Markov decision processes. *Mathematics of operations research*, 12(3):441–450, 1987.

[Pfeifer and Gómez, 2009] R. Pfeifer and G. Gómez. Morphological computation–connecting body, brain, and environment. In *Creating Brain-Like Intelligence*, pages 66–83. Springer, 2009.

[Quigley *et al.*, 2009] 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 *ICRA workshop on open source software*, 2009.

[Rickert *et al.*, 2014] M. Rickert, A. Sieverling, and O. Brock. Balancing exploration and exploitation in sampling-based motion planning. *IEEE Transactions on Robotics*, 30(6):1305–1317, 2014.

[Scholz *et al.*, 2014] J. Scholz, M. Levihn, C. Isbell, and D. Wingate. A physics-based model prior for object-oriented MDPs. In *Proc. of ICML-14*, pages 1089–1097, 2014.

[Schwaber, 2004] K. Schwaber. *Agile project management with Scrum*. Microsoft Press, 2004.

[Sucan and Chitta, 2016] I. I. Sucan and S. Chitta. MoveIt. `http://moveit.ros.org`, 2016.

[Thrun *et al.*, 2005] S. Thrun, W. Burgard, and D. Fox. *Probabilistic robotics*. MIT Press, 2005.

[van Gigch, 1991] J. P. van Gigch. *System Design Modeling and Metamodeling*. Springer Science & Business Media, 1991.

[Wolpert and Macready, 1997] D. H. Wolpert and W. G. Macready. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1):67–82, 1997.

[Wolpert, 1996] D. H. Wolpert. The lack of a priori distinctions between learning algorithms. *Neural computation*, 8(7):1341–1390, 1996.