

Interlocking Perception-Action Loops at Multiple Time Scales

A System Proposal for Manipulation in Uncertain and Dynamic Environments

Jeannette Bohg¹, Daniel Kappler^{1,3}, Franziska Meier^{1,2}, Nathan Ratliff³, Jim Mainprice¹, Jan Issac^{1,3},
Manuel Wüthrich¹, Cristina Garcia-Cifuentes¹, Vincent Berenz¹, Stefan Schaal^{1,2}

While grasping and manipulation in highly-controlled scenarios such as in factories is already possible, it remains unclear how a robot can achieve this autonomously in real-world scenarios that are characterized by a high degree of uncertainty. This uncertainty can be attributed to partial and noisy observations, a dynamic, constantly changing world and inaccurate actuation. While these factors are mitigated in industrial scenarios by introducing high amounts of structure, such simplifications cannot be leveraged in the domains like household or disaster relief scenarios where we would like to push robot operation.

A popular approach towards building an autonomous and robust manipulation system is through strong modularization according to the classic concept of *sense-plan-act*: The perception module provides the world model, in which a motion planner finds an optimal, often collision-free path that is then tracked by a stiff and accurate robot controller. Each module is expected to provide next to perfect solutions that can then be taken for granted by the subsequent modules. Information often flows only in one direction without feedback taken into account. This kind of approach is not robust against the challenging conditions present in the aforementioned scenarios. Recently, we have seen a more critical discussion of robotic system architectures as a result of multiple robotic challenges (e.g. the DARPA Robotics Challenge [1] or the Amazon Picking Challenge [2]) and the lessons learned from these. For example, in [1], the authors found that the areas of perception and autonomy pose one of the most difficult challenges. While many teams used existing software packages to solve different perception problems, they lead to mediocre performance when integrated in an entire system. This suggests that although these methods may perform well on isolated benchmarks, it is non-trivial to integrate them into an entire system. This may be due to the input being very different from what they have been tested on or different requirements on the robustness and noise-level of the output. In [2], the authors discuss four different axes that define a space in which robotic system architecture can be characterized. These axes stretch from (i) modularity to integration, (ii) generality to assumptions, (iii) computation to embodiment and (iv) planning to feedback. In this abstract, we propose an architecture that is mostly concerned with

integration instead of classic modularization and continuous feedback instead of planning.

For a robot to achieve manipulation behaviors under sensing and actuation uncertainty in dynamic environments, it needs to (i) continuously monitor the task-relevant parts of the environment including its own state, (ii) continuously re-plan to react to major changes in the environment and (iii) locally adapt the planned motion to cope with uncertainty and noise in the system.

Thus, we propose a system architecture that is organized in multiple, interlocked perception-action loops that run at different time scales. Each loop relies on some sensory feedback that may be provided at different rates and computes the next best motion plan or control input. At the center of our architecture is a motion optimizer that continuously optimizes robot arm motions given the current robot state, a target hand pose and some representation of the environment. The current robot state, the target object pose and the world representation are provided by real-time vision modules that use images from a depth camera as input. The resulting motion provides the reference for a compliant, low-level controller of the robot arm and fingers. Altogether, this system enables smooth and continuously adaptive motion generation in cluttered, dynamic environments for complex sequential manipulation tasks. Fig. 1 provides an overview of the proposed system.

Object and Manipulator Tracking: The system tracks simultaneously and in real-time both the object and manipulator state with respect to a depth camera that is mounted on the robot head. This provides a correct relative pose between object and end-effector independently of inaccuracies in hand-eye calibration. For object tracking, we rely on previous work described in [3]. It is robust to heavy occlusions, which are common in the context of object manipulation¹. For estimating the true robot arm configuration, we extend our previous work [4] by fusing online the joint measurements with the measurements from the depth camera. We deal with inaccuracies in the kinematics and the calibration by adding 6 virtual joints between the robot head and the camera. Both visual tracking methods run at the frame-rate of the camera [5].

World Modeling: When planning the robot motion, the system uses a geometric representation of the environment to avoid collision with obstacles and potential untracked objects or humans. This representation is essentially a signed

¹ Autonomous Motion Department at the Max Planck Institute for Intelligent Systems, Tübingen, Germany. Email: first.lastname@tuebingen.mpg.de

² Computational Learning and Motor Control lab at the University of Southern California, Los Angeles, CA, USA

³ Lula Robotics Inc., Seattle, WA, USA

¹Code available at <https://github.com/bayesian-object-tracking>

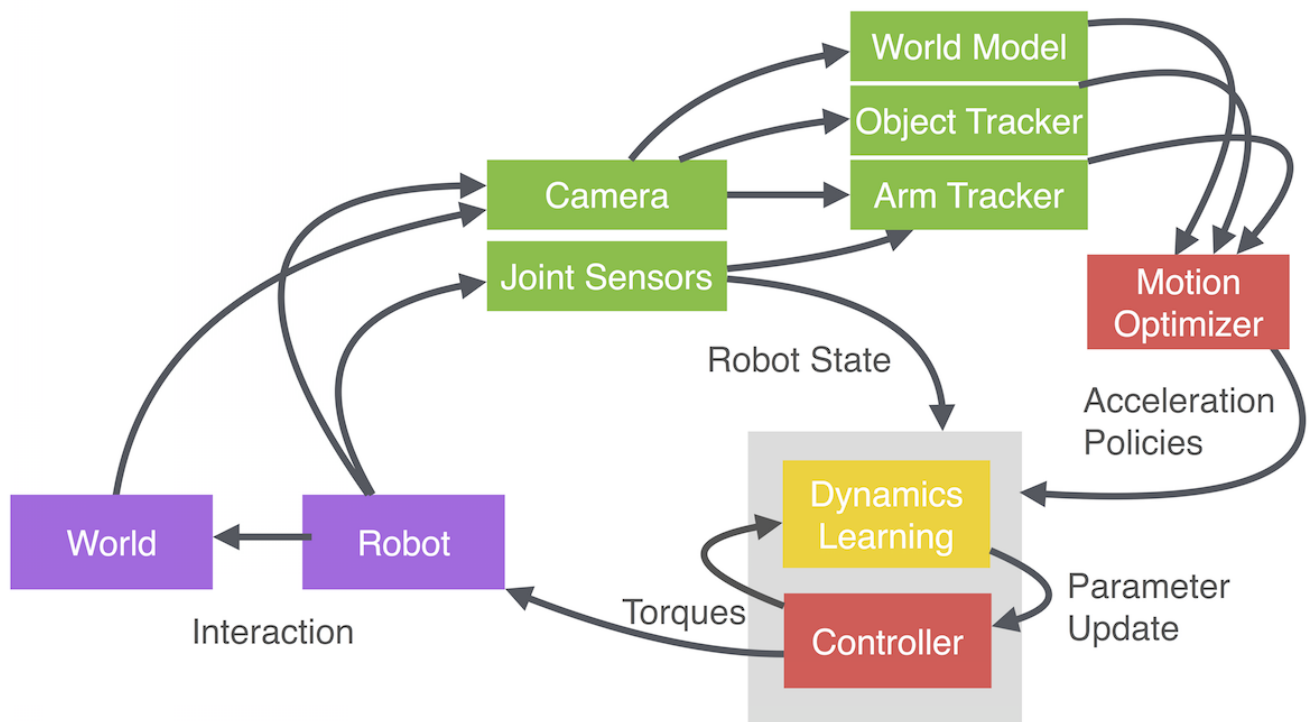


Fig. 1. Overview of the proposed system, its components and their interconnection. While each component provides a contribution within the specific field, this abstract focuses on the combination of those in multiple, interlocked perception-action loops which run at different time scales.

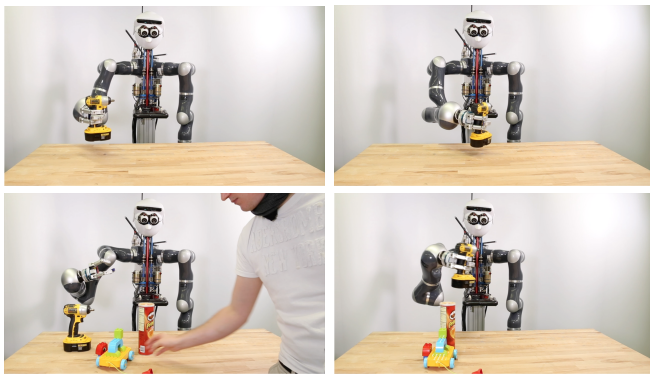


Fig. 2. The continuous motion optimizer to plan trajectories for reaching, grasping and placing the target object. Figures are showing an optimized motion in a simple (top) and more cluttered (bottom) environment. Each motion optimizes for the same target pose of the object.

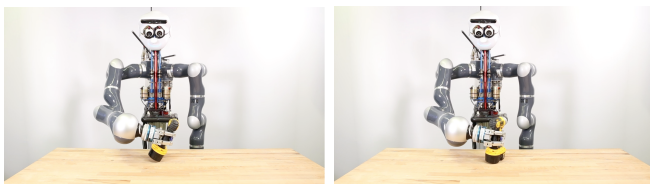


Fig. 3. Grasping a heavy object without (left) and with (right) dynamic model adaptation. Without the appropriate adaptation to the heavy payload, the low-level controller cannot track the desired trajectory.

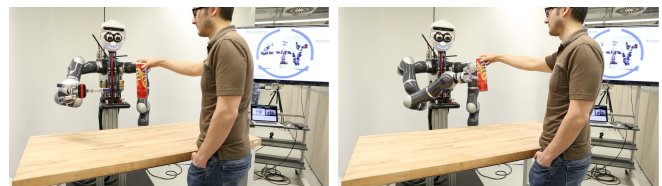


Fig. 4. Online optimization of the reaching trajectory to a dynamically moving object. This is enabled by the combination of the reactive planner and visual tracking methods for continuously monitoring the target object pose and current arm configuration.

distance function computed based on an occupancy voxel grid. This map is extracted from depth images, cropped to the robot workspace, removing the points corresponding to the robot arm and the tracked object, and setting occluded regions of space as occupied using ray casting. It is updated at a rate of approximately 20Hz. An extension of this representation can be found in [6].

Motion optimization: Since the state estimate of the robot and the target object configuration is constantly updated by the vision system, we need a motion generation module that is continuously adaptive. We use a multi-layered continuous optimization framework to generate and maintain a locally optimal motion policy for physically tracking and approaching the object even as the object moves. We decompose the grasp problem into multiple sequential task states—approach, establish grasp, move object, release, and retract—each governed by a separate finite-horizon motion optimization process. The currently active motion task takes

as input the robot's current state and the tracked relative object location, and re-optimizes the motion at a rate of approximately 10Hz using the Riemannian Motion Optimization (RieMO) framework [7]. The optimized motion policy is sent to control as a Linear Quadratic Regulator built from a second-order approximation to the optimizer's objective around the locally optimal trajectory; these policies are executed at the lower-level controllers rate of 1kHz in real time. Each task state optimizer sends a prediction of where it believes the policy will end up to the next task state optimizer in the sequence, so all task states maintain realistic policy predictions at all times that can be immediately executed upon transition into the state without the overhead of the initial policy optimization. This system enables smooth and continuously adaptive motion generation in cluttered environments for complex sequential manipulation problems.

Adaptive Inverse Dynamics Controller: Inverse dynamics control is a good framework to achieve precise and compliant tracking of acceleration policies generated by our continuous motion generation framework. Global accurate inverse dynamics models are difficult to obtain. Transparent integration of additional end effector payload or heat dependent system changes are even more complex to achieve. To this end, we propose combining inverse dynamics learning at different time scales, while achieving 1 kHz predictions for real-time control, to move towards a hybrid approach of *task-specific offline* and *task-agnostic online* modeling of the inverse dynamics errors [8].

Experiments and Demonstrations: We tested the system in three scenarios that demonstrate its general capability to adapt to complex, dynamic and uncertain environments. It achieves this through the combination of a reactive planner with continuous monitoring of the environment and robust, adaptive low-level control. As experimental platform we use the Apollo robot at the MPI for Intelligent Systems consisting of two Kuka LBR IV arms, two Barrett hands and an active humanoid head by Sarcos.

First, we show how the planner can adapt the optimized trajectory to a complex environment by taking the aforementioned world model into account. This experiment is illustrated in Fig. 2.

Secondly, we show how the low-level controller can adapt to heavy payloads when grasping objects. Despite an initially incorrect dynamics models, the robot can accurately track the desired trajectory after the adaptation. This experiment is illustrated in Fig. 3.

And lastly, we show how through a combination of visual object and arm tracking and continuous optimization of the

reaching trajectory, the system can cope with dynamic target objects. This experiment is illustrated in Fig. 4.

Discussion: We propose a system architecture that is capable of grasping and manipulation in uncertain and dynamic environment by heavily relying on fast feedback at different levels (planning and control) and integrating the different levels in perception-action loops instead of in a sequential architecture.

There are several points that could be addressed in the future. First of all in the current architecture, mostly visual and joint encoder feedback is taken into account. Manipulation tasks are however heavily concerned with contact interaction. This would benefit from taken haptic feedback from tactile arrays, strain gauges or force/torque sensors into account in the low-level controllers. Furthermore, the motion optimizer is currently mostly concerned with avoiding collision prior to grasping. It would be interesting to develop objective functions and motion policies that take contact interaction and exploitation of environmental constraints into account. These have been proven to provide the key towards robust grasping [9, 10, 11].

REFERENCES

- [1] C. Atkeson, B. Babu, N. Banerjee, D. Berenson, C. Bove, X. Cui, M. DeDonato, R. Du, S. Feng, P. Franklin, M. Gennert, J. Graff, P. He, A. Jaeger, J. Kim, K. Knödler, L. Li, C. Liu, X. Long, T. Padiar, F. Polido, G. Tighe, and X. Xinjilefu, "No falls, no resets: Reliable humanoid behaviour in the darpa robotics challenge," in *15th IEEE/RAS International Conference on Humanoid Robots*, 2015, pp. 623–630.
- [2] 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 robotic systems building," in *Robotics: Science and Systems Conference (RSS)*, June 2016, to appear.
- [3] M. Wüthrich, P. Pastor, M. Kalakrishnan, J. Bohg, and S. Schaal, "Probabilistic object tracking using a range camera," in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, 2013.
- [4] M. Wüthrich, J. Bohg, D. Kappler, P. C., and S. S., "The coordinate particle filter - a novel particle filter for high dimensional systems," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015.
- [5] C. G. Cifuentes, J. Issac, M. Wüthrich, S. Schaal, and J. Bohg, "Probabilistic articulated real-time tracking for robot manipulation," *RAS Letters*, 2016, submitted.
- [6] J. Mainprice, N. Ratliff, and S. Schaal, "Warping the workspace geometry with electric potentials for motion optimization of manipulation tasks," in *IEEE/RSJ Intl Conf on Intelligent Robots and Systems*, 2016.
- [7] N. Ratliff, M. Toussaint, and S. Schaal, "Understanding the geometry of workspace obstacles in motion optimization," in *IEEE Intl Conf on Robotics and Automation*. IEEE, 2015.
- [8] N. D. Ratliff, F. Meier, D. Kappler, and S. Schaal, "DOOMED: direct online optimization of modeling errors in dynamics," *CoRR*, vol. abs/1608.00309, 2016. [Online]. Available: <http://arxiv.org/abs/1608.00309>
- [9] L. Righetti, M. Kalakrishnan, P. Pastor, J. Binney, J. Kelly, R. Voorhies, G. Sukhatme, and S. Schaal, "An autonomous manipulation system based on force control and optimization," *Autonomous Robots*, vol. 36, no. 1-2, pp. 11–30, 2014.
- [10] M. Kazemi, J.-S. Valois, J. A. Bagnell, and N. S. Pollard, "Robust object grasping using force compliant motion primitives," in *Robotics: Science and Systems*, 2012.
- [11] R. Deimel, C. Eppner, J. Álvarez-Ruiz, M. Maertens, and O. Brock, "Exploitation of environmental constraints in human and robotic grasping," in *International Symposium on Robotics Research (ISRR)*, 2013.