Reactions and continuous adaptation in collaborative robots

Nathan Ratliff¹, Daniel Kappler¹,², Franziska Meier¹,²,³, Jan Issac¹,², Jim Mainprice², Manuel Wuthrich², Cristina Garcia-Cifuentes², Vincent Berenz², Dieter Fox⁴, Jeannette Bohg², Stefan Schaal²,³

I. INTRODUCTION

Collaborative robotics aims to bring humans and robots together to work as teams whose combined contributions and skills—superior perception and generalized dexterity in humans, and superior repeatability and precision in robots—surpass either party alone. But so far, collaborative robots in production environments have been largely agnostic of their surroundings, running through meticulously pre-programmed motions blindly, forcing their human collaborators to adapt and preventing the type of close proximity work required to fully leverage the benefits of human-robot teamwork. Perceptual systems improve at an aggressive pace, fueled by multidisciplinary incentives across both industry and academia. Developing new motion behavior and control technology that can anticipate and incrementally leverage the steady stream of advances in perception will have tremendous value, and can drive an important conversation around new research directions in vision that address issues around the innately sequential nature of active decision systems. Tight perceptual integration would enable safe close quarter collaboration in unstructured and uncertain environments while maintaining the precision and repeatability advantages of robotic systems.

We have developed a system for behavior generation from visual stimuli for this purpose, leveraging a multi-time-scale architecture combining continuous motion optimization with reactive local controllers. Our system turns the typically sequential sense-plan-act elements into a collection of parallel continuously running and continuously communicating processes. Vision provides a continuous stream of world state updates. That information is consumed by motion optimization, which continuously processes it and communicates desired behavior as a steady stream of reactive kinematic LQR policies to control, treating them as infinite collections of motion trajectories (their integral curves) rather than just a single trajectory. At the lowest level it combines these motion policies with other acceleration policies using quadratic programming, obviating the need to bend or otherwise adapt a reference trajectory to perturbations or environmental changes between re-optimizations.

We will be demonstrating the current state of our system live (in simulation) and use that to motivate discussions around further research, especially into issues around the integrating noisy and uncertain perception systems into physical real-time systems. See our demo video (https://youtu.be/DBic4vQ0Xs) for footage of the demos on the physical Apollo manipulation platform from the Max Planck Institute for Intelligent Systems.

II. ACCELERATION POLICIES AS TRAJECTORY BUNDLES

Motion optimizers and planners must operate under an assumed model of the system. However, unmodeled frictions, approximate dynamics, unpredictable end-effector loads, and other uncertainties make emulating the planned behavior on a physical platform challenging. So in practice it is important to have a kinematic reference signal that defines precisely the sequence of configurations the system should traverse to enable good controllers to generate the desired motion despite uncertainties. Most planners, for that reason, output one-dimensional kinematic trajectories through the configuration space. However, adapting these threads of execution and blending them between subsequent re-plans is challenging.

Our system, therefore, leverages second-order information around the local optimum of motion optimization problem to generate full acceleration policies of the form \( \ddot{q} = f(q, \dot{q}) \), which are much more expressive than a single trajectory since they effectively encode an infinite bundle of trajectories. Thus our system describes not only the planned behavior emanating from the state measured at the time of planning but also the behavior in an entire region around it.

A. Planning acceleration policies as kinematic LQRs

We use motion optimization to describe the motion generation problem as a constrained objective in a standard form described in [1], [2], and use Augmented Lagrangian [3] to rewrite the constrained problem as a series of unconstrained proxy objectives while estimating the Lagrange Multipliers. At convergence, we therefore have an unconstrained objective with penalty functions situated so as to satisfy the constrains, and we can calculate the Hessian around the local minimum of the objective. This Hessian and locally minimizing trajectory, in combination, form a time-varying positive definite local quadratic approximation of problem around the local optimum, which we can solve for the optimal time-varying affine policy using standard techniques from the control literature [4]. Because the original problem was defined purely on the kinematic level, this time-varying affine policy is an acceleration policy and therefore defines an entire bundle of system trajectories that a controller can follow.

Note that motion optimization is fundamentally slower than local control. That causes two issues in practice: 1. The system needs to start executing a re-optimized plan from an unexpected configuration since there is a time delay between when it starts the optimization process and when it finishes,
during which the underlying control system has progressed.

2. The system needs to react to perturbations and changes in the world faster than the motion optimization process can maintain.

The first issue is automatically resolved through the use of kinematic LQRs and time-varying affine acceleration policies. These policies can be executed from any robot state, so we just keep track of the time delay and start executing the new policy at the right time. Since the local minimum of the optimization process is positive definite and it is described purely on the kinematic level (without dynamics linearizations which commonly cause approximation problems in optimal control away from the linearization point), the integral curves are guaranteed to remain well-behaved and simply pull back toward a swath around planned region automatically if they deviate too far.

The next section discusses how combining the acceleration policy with local controllers can solve the second issue.

B. Combining with local control policies

Local control policies can be quite expressive, and in general we can describe them for collaborative robotic systems (or other such systems with invertible dynamics) as acceleration policies. For instance, one policy may prevent obstacle collisions, another may push away from self collisions and joint limits, and a third may maintain a desired end-effector target relative to a moving object. All of these can be combined with the optimized motion policy in a number of ways since they are all acceleration policies. The simplest technique is superposition of the differential equations (adding together the desired accelerations). That tends to cause suboptimal cancelling of the desired accelerations that slow down the overall system when too many policies are combined, so in practice we combine them through a standard local optimization similar to Quadratic Programming control techniques [5], [6].

III. LATENCY AND MULTIPLE TIME SCALES

The world changes faster than motion planners can re-optimize, and perception systems provide visual stimuli often at around 30hz, or even faster when considering other forms of sensory input. Combining motion optimization, which operates at between 10hz and 1hz (depending on the difficulty of the (re-)optimization problem), and very fast local acceleration policies, which can operate at around 1khz, lends itself well to real-time adaptation and reactions to the changing environment. Our system remains locally safe and actively avoids collisions through even severe perturbation, while continuously adapting its longer term motion policy to retain the forward-looking benefits of motion optimization even through substantial changes to the problem.

IV. INTEGRATION WITH VISION

Our system is designed to leverage visual stimuli of many forms, including future improvements as research in the field progresses. From fully motion captured human movement in a research setting, to overhead cameras viewing well-known objects on a table of pre-specified height, to fiducials, to structured or (eventually) unstructured settings leveraging 3D depth cameras or trained tools for estimating structure directly from 2D images, there is consistently a need to integrate that information into the motion generation system. The question of what we do with that information can already be addressed now.

We have integrated a visual system into our setup on the physical Apollo platform at the Max Planck Institute for Intelligent Systems to demonstrate closing the loop between visual stimuli and reactionary responses. The visual 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 [7]. It is robust to heavy occlusions, which are common in the context of object manipulation (code available at https://github.com/bayesian-object-tracking). For estimating the true robot arm configuration, we extend our previous work [8] by fusing online the joint measurements with the measurements from the depth camera (see https://youtu.be/ESTgkCqdP1I).

V. CONCLUSIONS AND FUTURE WORK

Collaborative robots must be fundamentally adaptive to their surroundings, which means handling visual stimuli continuously and responsively. We present a new architecture for continuous motion planning and control that simultaneously maintains real-time reactivity while adapting the planned motion using computationally more expensive but substantially more expressive motion optimization tools. An important avenue of future work is in the vision integration. Excellent vision tools have been demonstrated in research settings (see [9], [10], [11]) but there remain open challenges, especially in initialization, robustness to substantial occlusions, data association, leveraging sequential observations, and integrating other sensor modalities such as force or tactile feedback.

REFERENCES