

Experimental Validation of Reference Spreading for Robotic Manipulation of Unmodeled Objects

Gijs van den Brandt

Abstract—Exploiting impacts can speed up robotic object manipulation. However, when using traditional control approaches, impacts are paired with peaks in joint torques and contact forces that may damage the robot or its environment. Reference spreading (RS) is an impact-aware control approach that addresses these peaks without significantly compromising on tracking performance. Numerical studies have validated the benefits of RS controllers for object manipulation tasks, but these controllers relied on knowledge regarding the robot's contact state and accurate object models – information that is not easily available in a real-life setup.

This work describes an impact detection algorithm and a model-free trajectory planner using teleoperation. Furthermore, a soft end effector is designed. These components are implemented on a dual-arm robotic setup to evaluate RS in a usecase that is relevant to industry, namely grabbing a box. Experiments show that RS reduces control effort – and therefore joint torques and contact forces – without inhibiting tracking performance. This means that impacts with higher velocities become feasible, and subsequently that RS can contribute to faster object manipulation.

I. INTRODUCTION

Automation has historically played a crucial role in the logistics industry. Our current way of living depends on autonomous systems for global transportation and warehousing. The growing labor shortage and increasing demand for online retail motivate further developments in the logistics sector [1].

An example of automation in logistics can be found in depalletizing, where object are removed from a pallet. Three different depalletizing approaches can be found in industry.

A logistical aspect where machines struggle to fully compete with humans is object manipulation. A practical examples of this is depalletizing. While robots are strong and consistent when manipulating objects, humans are versatile and swift. Robots are held back from faster performance because they must often slow down prior to making contact; establishing contact at a high velocity – an event referred to as an impact – could cause damage to the robot or its environment. On the contrary, humans intrinsically exploit impacts in the form of grabbing, bouncing and hitting. (give better explanation of depalletizing: manual and automatic approach)

The field of impact-aware control aims to better equip robots for making contact at high velocities. These impacts are paired with large contact forces that could damage the system. Previous work describes control using the maximum allowable impact velocity that complies with safety constraints such as limits for the contact force [2, 3]. This was combined with a compliant cover for the robot that

reduces contact forces at impact, facilitating higher feasible impact velocities. Rather than using a soft cover, compliancy may also be achieved by designing a robot with low inertia and high backdrivability as was done in [4].

In addition to the large contact forces, a subject of interest is the velocity jump at the time of impact. Time misalignments between velocity jumps in the reference and in the actual system cause the velocity tracking error to peak[5], as is shown in Figure xxx. This error peak results in undesired control effort and should therefore be avoided. In [6], the robot's velocities are projected into an impact-invariant subspace based on the expected point of impact. As a result, impact-driven peaks in the velocity tracking error are reduced significantly. It is not always possible to describe a point of impact, however. Often times, impacts occur between surfaces rather than just points. Furthermore, corners of the surface may impact at diverging intervals in uncertain order during what is called near-simultaneous impacts, shown in Figure xxx.

The impact-aware control scheme called Reference Spreading [7] addresses error peaking caused by misaligned impacts. It operates on the basis of a tracking error that switches once an impact is detected. This concept is best explained at the hand of Figure xxx. The reference is split at the nominal impact time into an ante- and post impact reference. These references are then extended. Initially, the tracking error is based on the extended ante-impact reference, but this is switched to the post-impact reference once an impact is detected. Evidently, this can reduce the error peaking.

Reference spreading can also handle simultaneous impacts. [8] (explanation)

By addressing the peaking error, reference spreading facilitates faster object manipulation, making it interesting to industry if its effectivity can be proven in practice. Reference spreading for object manipulation has already been validated in simulations [8, 9]. Experimental validations have been limited to interaction with a fixed environment, however [10, 11]. The goal of this work is therefore to **provide a real-world implementation of reference spreading for practical object manipulation tasks**. To translate the results from simulation to reality, the following contributions are made:

1. Motion planning for impacts without object models: Generating a reference with velocity jumps that is coherent with the systems dynamics is challenging. One approach maps the ante-impact velocity to the post-impact velocity based on conservation of momentum [ref for impact map].

This requires a model of the environment, which is feasible in simulations with simplified dynamics, but challenging in reality. Impact-driven velocity jumps could instead be inferred experimentally. In previous studies [12], the control gains are reduced to zero upon detection of the impact while inferring an impact map, so that the velocity jump would not result in excessive motor torques. A different model-free motion planning strategy is proposed, which not only produces velocity-reference jumps that are coherent with the system dynamics, but also leverages human intuition to generate fluid motions before and after the impact. This is achieved by introducing a human in the loop by means of teleoperation.

2. Impact detection: The reference spreading scheme should switch between ante-, intermediate-, and post-impact references at the appropriate time. This requires an impact detection algorithm. Approaches in literature look either at position data [13] or external force estimations [11, 14, 15] for signs that could be caused by impacts. We show that these signs are necessary, but not sufficient conditions for an impact – only looking at position or contact force can result in false positives. To limit false detection of impacts, a novel impact detector that looks at both force and position data is proposed and evaluated.

3. Custom end effector:

(4. Intermediate impact phase controller:)

II. BACKGROUND

A. Robots suitable for impacts

goal: Explain why franka emika robot is suitable for impacts, explain torque control, show custom end effectors from literature, motivate design of a new end effector

Traditionally, robots are designed to be stiff.

B. Quadratic programming control

goal: motivate QP control, formulate cartesian impedance task, posture task, and list safety constraints

C. Reference spreading

D. Impact detection

III. TRAJECTORY PLANNING

IV. METHODS

A. Soft end effector

B. Trajectory planning via teleoperation

C. Reference spreading formulation

V. EXPERIMENTAL VALIDATION

A. Implementation

B. End effector evaluation

C. Reference spreading

D. Intermediate modes

VI. CONCLUSION

APPENDIX

- A. Rotation mathematics*
 - B. End effector design*
 - C. Supplementary experimental results*
- hoi