Hochschule
Bonn-Rhein-Sieg
University of Applied Sciences

b-it Bonn-Aachen International Center for Information Technology

R&D Project

Exploiting contact constraints in robotic manipulation tasks

Wing Ki Lau

Submitted to Hochschule Bonn-Rhein-Sieg,
Department of Computer Science
in partial fullfilment of the requirements for the degree
of Master of Science in Autonomous Systems

Supervised by

Prof. Dr. Nico Hochgeschwender
Sven Schneider, M.Sc.

August 2023

I, the undersigned below, declare that this work has not previously been submitted to this or any other university and that it is, unless otherwise stated, entirely my own work.

____
Date

____
Wing Ki Lau

Abstract

Your abstract

Acknowledgements

Thanks to ....

Contents

1

Introduction

In the field of industrial robotics, achieving high levels of accuracy, precision, and repeatability in free space motion is a critical requirement. To meet this requirement, robots must be designed to be stiff and avoid contacts along their trajectories. One approach to achieving a high degree of stiffness is to use lightweight materials such as carbon fiber. However, the production of carbon fiber is expensive, leading some companies to opt for cheaper and heavier robot manipulators to reduce manufacturing costs. While this approach may reduce costs, it also results in increased energy consumption as the robot must expend more energy to maintain its joints and links in mid-air.

During motion planning, robot manipulators commonly treat contact surfaces as obstacles rather than opportunities. This approach contrasts with the behavior of humans, who often rest their wrists on a table while writing on paper. By exploiting contact surfaces, robots can achieve selective improvements in accuracy and precision compared to writing on paper with their wrists in mid-air. Furthermore, utilizing contact surfaces can help reduce energy consumption.

To enable robots to imitate this behavior, two adaptations of control software are required: Dynamics and partial constraint. Contact handling requires not only the kinematics of the robot but also its dynamics. There are existing dynamic solvers that can be used to solve the robot's equations of motion. To fully exploit the contact surface, the specification of partial constraints must be introduced. The acceleration-constrained hybrid dynamics algorithm (ACHDA) limits the ability to handle the partial constraint of an end-effector. ACHDA can handle partial motion specifications in some direction and leave them determined by nature. Since nature determines part of the motion, the solver provides higher energy efficiency, passive adaptation, and desire constraints that do not require explicit control. Shakhimardanov has demonstrated the potential for extending constraints to arbitrary links [1]. However, this extension has not yet been incorporated into any software library. This means that while the theoretical framework for extending constraints to arbitrary links has been established, it has not yet been implemented in practice

In this research and development project, we will address three case studies to demonstrate a robot that exploits environmental constraints: Grasping an object by sliding along its surface, performing a writing task, and resting robot manipulation. The first case study aims to establish the required infrastructure for task execution through a proof-of-concept task. The second case study aims to implement the extension of the Vereshchagin solver in daily tasks to investigate whether exploiting contact surfaces can improve the accuracy and precision of robot motion. The third case study aims to evaluate whether resting some of the joints on a supporting surface can lead to higher energy efficiency.

The content of this research and development report is presented as follows. (Add after the meeting when the overall structure is fixed)

2

Problem Statement

Many robots today are heavy, energy-consuming, and costly to manufacture. During motion planning, contact surfaces are often treated as obstacles, resulting in the robot's joints remaining in mid-air. This approach can lead to wasted energy as the robot must expend energy to maintain its joints and links in such a pose. Additionally, existing dynamic solvers do not handle partial constraint conditions. In this R&D project, we will explore an approach that addresses the dynamics of this situation. By considering contact surfaces as opportunities rather than obstacles, we aim to develop a more energy-efficient and cost-effective solution for robotic motion planning.

3

State of the Art

3.1 Robot dynamic

A robot manipulator, refers to a mechanical system consisting of a set of rigid bodies call *links* which interconnected by *joints*. These joints facilitate relative motion between neighboring links, granting the manipulator the capability to maneuver in a variety of ways within its workspace. The spatial arrangement of the manipulator's joints and links lead to physical constraint which limits the direction of the motion that is being executed.

The motion of a rigid body, to be precise, the forces and accelerations of a rigid body, are being studied in the field of robot dynamics which encompasses both *forward* and *inverse* dynamics. The motions are described by the dynamic equation which being evaluated by dynamic algorithm which solve the numerical value that related to the dynamics [2]. In *forward Dynamic*(FD), an applied force to a rigid body is given to calculate the acceleration respon of such input [2]. The equation of motion of FD is expressed in the following form:

$$\ddot{q} = H(q)^{-1}(\tau - C(q, \dot{q})) \tag{3.1}$$

In equation (3.1), τ is a vector of generalized force in joint space. q , $\dot{q}$ and $\ddot{q}$ are the vector of position, velocity and acceleration in joint space respectively. $H(q)$ is the interia matrix which is a function of q. At last, $C(q, \dot{q})$ represent the generalized bias force which comprises Coriolis, centrifugal and gravity forces in joint space with other external force that possibly act on the system [2].

Conversely, Inverse dynamics (ID) computes the necessary force required to generate a given or desired acceleration within a system of rigid bodies. The equation of motion of ID is expressed in the following form [2]:

$$\tau = H(q)\ddot{q} + C(q, \dot{q}) \tag{3.2}$$

Besides forward dynamic and inverse dynamic, Hybrid Dynamic (HD) contains both dynamic problems by given/known τ or $\ddot{q}$ of a specific joint to compute the unknown forces and accelerations.

3.2 Dynamic solver

This session introduces some common dynamic algorithms that being used to solve the dynamic problem mentioned aforenamedly.

1. The Articulated Body Algorithm (ABA) computes the dynamics and motion of articulated bodies. This algorithm models the dynamics of each body in the system using a hierarchical approach, taking into account the interactions between the bodies and the forces acting upon them. The ABA solver is used to solve a forward dynamics problem, where accelerations are calculated from input torques [3].

2. The Recursive Newton-Euler Algorithm (RNEA) is a recursive approach to solving the inverse dynamics problem, which involves calculating joint-space torques from joint-space accelerations for each body in a system. This algorithm takes into account the relationships between the bodies and the forces acting upon them. It is aimed to streamline and improve the computation of dynamics in complex multi-segmented robot manipulator. The recursive approach minimizes repetitive calculations and boosts computational effectiveness. [2].

3. The Articulated-Body Hybrid Dynamics Algorithm (ACHDA) is capable of computing three types of dynamics: inverse dynamics, forward dynamics, and hybrid dynamics. In inverse dynamics, the solver calculates joint torques from Cartesian acceleration constraints. In forward dynamics, the solver takes feed-forward joint torques as input and computes Cartesian accelerations. In hybrid dynamics, the solver combines both of the above methods. Additionally, as a byproduct, it can also take external forces as input and compute joint accelerations. [2].

As a safer method to visualise the effect of validating the solution of control signal and power consumption,in research and development,researchers and roboticitsts frequently use stimulation to perform experiments and developments before deploying on a physical robot. There are many existing software solutions and libraries provide physics engine solutions for simulating physical interactions and dynamics in robotics. Dynamic Animation and Robotics Toolkit (DART) [4], Open Dynamics Engine (ODE) [5], Bullet Real-Time Physics engine [6] and MuJoCo (Multi-Joint dynamics with Contact) physics engine [7]. The physics engines menntioned above uses numbers of aformentioned dynamic solvers such as The Recursive Newton-Euler Algorithm (RNEA) and Articulated Body Algorithm (ABA).

3.3 Controller

Controller guides systems toward desired setpoints or references value with feeback loop. The followings are some common controllers that are widely used in rocotics.

3.3.1 Proportional-integral-derivative (PID) controllers

Proportional-Integral-Derivative (PID) controllers uphold a specific setpoint. In this R&D project, PID controller regulates position and velocity by using the equation below:

$$u(t) = K_p e(t) + K_i \sum_{t=0}^{t} e(t)\Delta t + K_d \frac{e(t) - e(t-1)}{\Delta t} \tag{3.3}$$

Where $u(t)$ is the control signal at time t. $e(t)$ is the error at time t as the difference between the setpoint and the current input of the controller. Δt is the time step of each calculation in the control loop. K_p, K_i, K_d are the coefficients of proportional, integral and derivative gain respectively [8].

3.3.2 Fuzzy logic algorithm

Fuzzy logic algorithms employed as controllers adeptly navigate scenarios marked by uncertainty, adeptly utilizing linguistic terms and membership functions to accurately represent the gamut of inputs and outputs. This approach hinges upon rule-based decision-making, whereby the system gauges the extent of each rule's applicability contingent upon the values presented in the input domain [9].

3.3.3 Impedance controllers

Impedance controller emulate the the mechanical properties of stiffness and damping to regulate the forces and motions relationship when the robot interacts with objects. Extend from equation (3.2) with two main components: Stiffness and Damping Component. A basic Impedance controller is formed.

$$\tau = K(q_d - q) + D(\dot{q}_d - \dot{q}) + H(q)\ddot{q} + C(q, \dot{q}) \tag{3.4}$$

Where K and D are stiffness-like and damping-like matrix, respectively [10].

3.4 Task based control

Task-based control in robotics refers to a control approach that focuses on achieving specific tasks or objectives rather than directly controlling individual joints or degrees of freedom. Instead of commanding the robot's joints to move in a prescribed manner, task-based control involves specifying the desired behaviors or goals the robot should accomplish.

3.4.1 iTaSC

iTaSC is a control framework that was introduced by [11]. It is the fusion of instantaneous task specification and estimation of geometric uncertainty that focuses on achieving tasks while respecting various constraints. These constraints could include safety considerations, physical limitations, interaction forces, and more. The framework provides a way to dynamically adjust a robot's behavior in real time,

allowing it to respond to changing conditions and maintain safe and effective interactions with the environment. Later [12] extended the idea of iTaSC.
In a multi-sensor robot system, a task is described as the desired behaviors or objectives that the robot needs to achieve. These tasks are formulated in the task space, which represents the robot's end-effector or operational space. The tasks and constraints had to be formulated at the begining. Task formulation encompasses goal position, following a particular trajectory or maintaining a certain force. constraint formulation contains obeying joint limits, velocity limits ,avoiding collisions. etc. Based on their priorities,tasks and constraints are organized in a hierarchical manner. Two set of coordinates were introduced to represent the tasks and constraints, feature and uncertainty coordinate. Feature coordinate expresses the relative mottions between feature on the specific obejcts. uncertainty coordinate represent geometric uncertainties.
The authers proposed general control and estimation scheme: Where the robot system and the environment are are denoted as *plant* P which observer the senser measurements z. $u$is the control input (joint posiitons,velocities and torques) that compute from controller C. Contoller takes the estimates X_u and $\widehat{y}$ as input where both of them are generated from *Model Update and Estimation block* M+E estimate $\widehat{y}$ which takes the controlled input signal u and measurements z as input to estimate y as the output of the system.

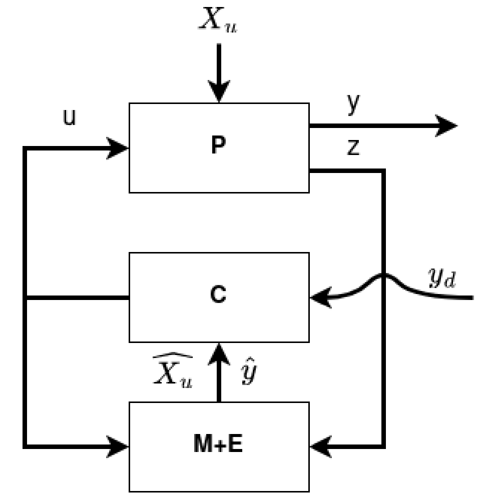

Figure 3.1: A general control and estimation scheme. [11]

Later, [13] poposed using iTaSC on force-sensorless robot force control. Figure 3.2 show the one degree-of-freedom (DOF) robot force control scheme that can extend to mult degree-of-freedom scenerio.

In figure 3.2, the robot system model and the underlying joint velocity controller are depicted in red. The stiffness model pertaining to environmental contact is showcased in green, while the controller section open to design is highlighted in blue. The gravity term is omitted because the robot was being used in the experiment was gravity-compensated mechanically. This is a first order system with a desired input τ_d

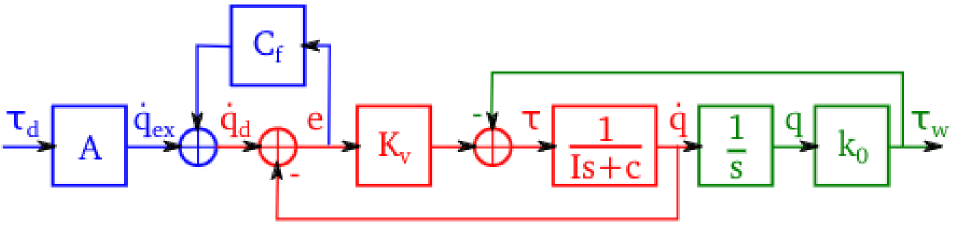

Figure 3.2: One degree-of-freedom robot force control scheme. [13]

and output $\dot{q}$ in non-contact situation. In contrast,the green part is included. The model characterizes a second-order system where τ_d serves as the input, and the resulting torque τ_w exerted on the environment acts as the output. The joint velocity error feedback constant, C_f, results in the modification of the velocity loop feedback constant K_v, leading to an equated velocity loop feedback constant $\frac{K_v}{1-C_f}$. In mult degree-of-freedom scenerio, all the previously mentioned variables becomes vector and the constants becomes matrices [13].

3.4.2 Stack of tasks

The paper from [14] and [15] introduced Stack of tasks (SoT) which provides a way to manage multiple objectives and constraints in a coherent and organized manner. It ensures that the robot's control algorithm focuses on achieving the most critical tasks first while considering the constraints and interactions among tasks and generalized the concept of hierarchy-based control strategies to include unilateral constraints at various levels of priority.

3.5 Existing implementation of exploting surface in robot manipulation

There were some existing implementation or discussion about exploting surface in robot manipulation. In [16] ,authors presented a mathematical framework that outlines the interaction between compliant hands and environmental constraints during the execution of a grasping task. [17] discussed planning grasp strategies that exploit environmental constraints. In this paper, each environmental constraint exploitation is considered as one controller with a desired spatial and contact condition with a termination predicate which is also a switching condition if the constraint is in between the motion in global point of view. The final motion planing consists of a series of environmental exploits that lead to a grasp.

3.6 Limitations of previous work

In today's world, there's an ongoing challenge in deciding between investing in an expensive, lightweight robot arm or a more affordable but heavier one. This decision-making is influenced by limited financial resources for development. Opting for cheaper, heavier robots can lead to smoother but less precise robotic movements, affecting accuracy.

Interestingly, current robot motion planning still treats contact surfaces as obstacles to be avoided. The well-known dynamic solvers like ABA and RNEA don't handle situations where only some constraints are present very well. As of now, the existing researches relys on visual or other sensory input to detect contact surface. However,there's a lack of research on how to adapt the Vereshchagin solver to scenarios where contact is made by joints other than the end-effector, and how to practically implement this idea without including sensors on the robot.

4

Background

In section 3.6, the limitations of the current research has been raised. In order to control the robot to move along a contact surface without sensory input, there were researchers, Vereshchagin and Popov ,proposed a hybrid dynamic algorithms and tried to tackle the problem of exploting the defined contraints,robot model, joint angles, joint velocities, feed forward torque and external forces to compute the required joint torque and joint accelerations as command that being sent to the robots [18].

4.1 Active contraint matrix and acceleration energy setpoints

4.2 Calculate differences with KDL library

5

Solution

5.1 Proposed robot archtechture

5.2 Extensin of Robif2b

6

Evaluation and results

In this RnD, two experiements are being conducted with respected to two use cases: (1) Grasp object by sliding motion along surface and (2) Perform writing task. In this chapter, the methodology will be divided according to use cases.

6.1 Use case 1 - Grasp object by sliding motion along surface

The aim of this case study is to grasp the object successfully by sliding the robot manipulator along a contact surface. Assume that the object pose is known, only the object to be grasped is on the table , no obstacle and the contact surface is known. The manipulator should first approach the contact surface, for example, a table with the target object placed on it. When the manipulator is above the contact surface, it moves toward the surface until establishing contact. By activly monitoring the velocity along linear Z axis v_{lin_z}in world frame. If the absolute value of v_{lin_z} for 10 samples is less than a threshold value. The contact between a surface and the robot manipulator is being established. After contact is established, the manipulator slides along the linear x direction for 10 cm $d_x = 0.1$ until it reaches a grasping region. Finally, the end-effector performs a grasping motion.

6.1.1 Setup

- Kinova® Gen3 Ultra lightweight robot maniputlator is attached to a table
- The motion will starts at a fixed starting pose $q = 6.28318, 0.261895, 3.14159, 4.01417, 0, 0.959856, 1.57079$ in radian

6.1.2 Experimental Design

-

6.2 Use case 2 - Perform writing task

The aim is to draw a line on the paper with a wrist joint contacting the writing surface like human writing. Before starting the manipulation task, the gripper firmly grasps the pen or marker. The manipulator should first approach the contact surface, for example, a table with the target object placed on it. When the manipulator is above the contact surface, it moves toward the surface until establishing contact. Once contact is established, the manipulator draws a line according to a predefined motion specification. The evaluation will be a trajectory comparison in terms of position or velocity with and without contact between the robot and the support surface

6.2.1 Setup

6.2.2 Experimental Design

-

6.3 Use case 3 - Resting elbow manipulation

6.3.1 Experimental Design

-

7

Conclusions

7.1 Contributions

7.2 Lessons learned

7.3 Future work

A

Coordinate transformation for force and motion vectors

p.22 from featherstone

B

Plucker coordinate for spatial vectors

C

Cross product operators

D

Parameters

References

[1] A. Shakhimardanov, “Composable robot motion stack,” Ph.D. dissertation, 2015.

[2] R. Featherstone, *Rigid Body Dynamics Algorithms.* Berlin, Heidelberg: Springer-Verlag, 2007.

[3] ——, “A divide-and-conquer articulated-body algorithm for parallel o (log (n)) calculation of rigid-body dynamics. part 1: Basic algorithm,” *The International Journal of Robotics Research*, vol. 18, no. 9, pp. 867–875, 1999.

[4] J. Lee, M. X. Grey, S. Ha, T. Kunz, S. Jain, Y. Ye, S. S. Srinivasa, M. Stilman, and C. K. Liu, “DART: Dynamic animation and robotics toolkit,” *The Journal of Open Source Software*, vol. 3, no. 22, p. 500, Feb 2018. [Online]. Available: https://doi.org/10.21105/joss.00500

[5] [Online]. Available: http://www.ode.org/

[6] Admin, “Bullet real-time physics simulation,” Mar 2022. [Online]. Available: https://pybullet.org/wordpress/

[7] [Online]. Available: https://mujoco.org/

[8] M. A. Johnson and M. H. Moradi, *PID control.* Springer, 2005.

[9] E. Dadios, *Fuzzy logic: algorithms, techniques and implementations*, 2012.

[10] P. Song, Y. Yu, and X. Zhang, “Impedance control of robots: an overview,” in *2017 2nd international conference on cybernetics, robotics and control (CRC).* IEEE, 2017, pp. 51–55.

[11] R. Smits, T. De Laet, K. Claes, H. Bruyninckx, and J. De Schutter, “itasc: a tool for multi-sensor integration in robot manipulation,” in *2008 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems.* IEEE, 2008, pp. 426–433.

[12] W. Decré, R. Smits, H. Bruyninckx, and J. De Schutter, “Extending itasc to support inequality constraints and non-instantaneous task specification,” in *2009 IEEE International Conference on Robotics and Automation.* IEEE, 2009, pp. 964–971.

[13] D. Vanthienen, S. Robyns, E. Aertbeliën, and J. De Schutter, “Force-sensorless robot force control within the instantaneous task specification and estimation (itasc) framework,” in *Benelux Meeting on Systems and Control*, 2013.

[14] N. Mansard, O. Khatib, and A. Kheddar, “A unified approach to integrate unilateral constraints in the stack of tasks,” *IEEE Transactions on Robotics*, vol. 25, no. 3, pp. 670–685, 2009.

[15] N. Mansard, O. Stasse, P. Evrard, and A. Kheddar, "A versatile generalized inverted kinematics implementation for collaborative working humanoid robots: The stack of tasks," in *2009 International conference on advanced robotics*. IEEE, 2009, pp. 1–6.

[16] G. Salvietti, M. Malvezzi, G. Gioioso, and D. Prattichizzo, "Modeling compliant grasps exploiting environmental constraints," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 4941–4946.

[17] C. Eppner and O. Brock, "Planning grasp strategies that exploit environmental constraints," in *2015 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2015, pp. 4947–4952.

[18] A. F. Vereshchagin, "Modeling and control of motion of manipulational robots," *Soviet journal of computer and systems sciences*, vol. 27, no. 5, pp. 29–38, 1989.