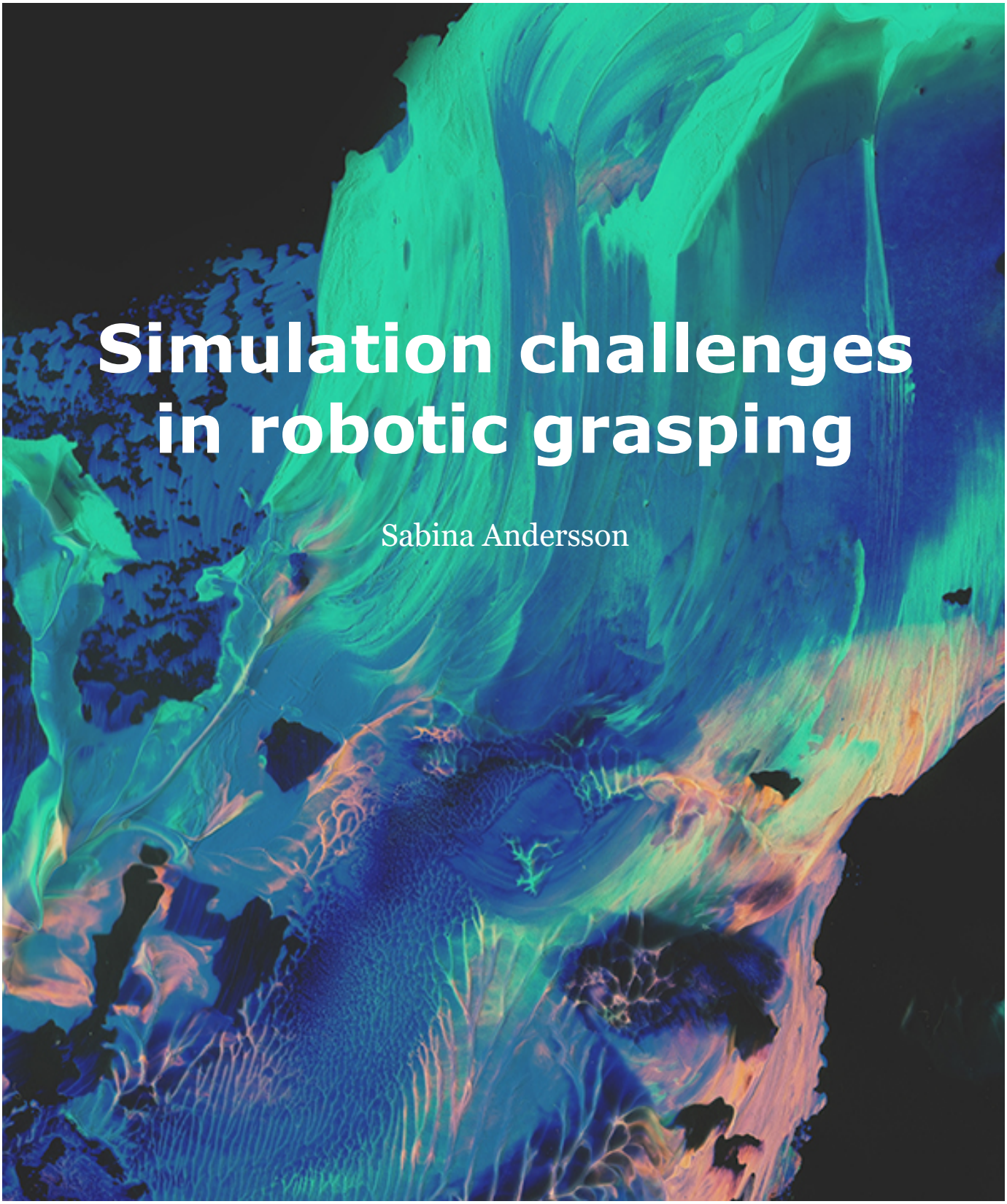
UMEÅ UNIVERSITY



Simulation challenges in robotic grasping

Sabina Andersson

Master's thesis in Engineering Physics, 30 ECTS
Department of Physics, Umeå University, Spring 2021

Simulation challenges in robotic grasping

Master's thesis in Engineering Physics
Department of Physics, Umeå University
June 2021

Author

Sabina Andersson
sabinacharlotta@gmail.com

Place for Project

Algoryx Simulation AB
Umeå, Sweden

Examiner

Eddie Wadbro, Department of Computing Science
Umeå University

Supervisors

Martin Servin and Yuewei Ma

Algoryx Simulation AB

Abstract

Grasping and dexterous manipulation is a huge area in the current robotic research field. Traditionally in industrial environments, robots are customized for a certain task and work well with repetitive movements where the entire process is predetermined and deterministic. The possibility for a robot to adapt to its surroundings and manipulate objects with unknown properties is very limited and requires new models and methods. It is advantageous to explore and test new algorithms and models in a simulated environment before building physical systems. This thesis focuses on making the development, design and control of dexterous robot hands easier by exploring the benefits and challenges to use simulation in Algoryx's physics engine AGX Dynamics for experimentation and evaluation of design, motion planning, contact models, geometry, etc. The goal is to simulate complex grasping situations in AGX Dynamics and to build knowledge about contact mechanics to fully capture the dynamics of grasping in simulation. The work consists of validating the simulation library with regards to fundamental physics characteristics involved in grasping by a series of benchmark tests. The validation process aims to verify the models and numerical methods in AGX Dynamics to ensure sim2real transfer. The work has exposed friction as one of the biggest challenges in simulation. Developments of the current friction models that better represent the reality have been implemented and tested in AGX. They both show promising results for further development.

Keywords

Robotic grasping, Friction models, Sim2Real, Validation

Abstract

Greppning och fingerfärdig manipulering av object är just nu ett stort forskningsområde inom robotik. Inom industrin är robotar traditionellt skräddarsydda för en specifik uppgift och fungerar väl med repetitiva uppgifter där hela processen är på förhand bestämd. Möjligheten för en robot att anpassa sig till sin omgivning och hantera objekt med okända fysikaliska egenskaper är mycket begränsad och kräver nya modeller. Det är fördelaktigt att utveckla och testa nya algoritmer och modeller i en simulerad miljö innan man bygger fysikaliska system. Den här uppsatsen fokuserar på att göra utveckling, desing och kontroll av fingerfärdiga robothänder enklare genom att undersöka fördelar och utmaningar med att använda Algoryx fysikmotor AGX Dynamics till utveckling och utvärdering av design, rörelseplanering, kontaktmodeller, geometri, etc. Målet är simuleringar av komplexa greppnings-situationer i AGX Dynamics och att bygga kunskap kring kontaktmekanik för att i simulering fånga dynamiken i greppning. Arbetet består av validering av simuleringsbiblioteket med fokus på fysik involverat i greppning genom flera benchmark-test. Valideringsprocessen siktar mot att verifiera modeller och numeriska metoder i AGX Dynamics för att säkerställa överföring från simulering till verklighet. Arbetet har visat att friktion är en av de största utmaningarna i simulering. En utveckling av de nuvarande friktionsmodellerna, som bättre motsvarar verkligheten, har blivit implementerade och testade och visar goda resultat för vidare utveckling.

Nyckelord

Robotgreppning, Friktionsmodeller, Validering

Acknowledgements

I would like to send my deepest thanks to my supervisors Yuewei Ma and Martin Servin at Algoryx for always being helpful, supportive and positive during the time of my thesis work. Special thanks to Claude Lacoursière for sharing your knowledge with me and the great banter. Thank you to Niklas Melin and Tomas Berglund for building and helping with the implementation of our new ideas. To the rest of the wonderful people at Algoryx: Thank you for including me and always finding an answer to my many questions. Lastly and most importantly, huge thanks to 'Eliten' for these amazing five years we've had together. You guys are the reason for why we made it.

Acronyms

PGS	projected Gauss-Seidel
ODE	ordinary differential equation
DAE	differential algebraic equation
LCP	linear complementarity problem
NCP	nonlinear complementarity problem
MLCP	mixed linear complementarity problem
ADR	automatic domain randomization
sim2real	simulation to reality

Contents

Chapter 1

Introduction

The use of robots in manufacturing industries has been extensive over the last few decades. Robots are extremely well suited for taking on repetitive tasks which can streamline the overall assembly workflow and does not get tired and make mistakes in the same way as humans do. Industrial robots today are usually tailor-made for a certain task with predetermined movement patterns and generally have reduced ability to adapt to their surroundings. They are most effective with repetitive familiar settings where geometry, weight, and material properties of the objects they are handling are known in advance.

The current trend is moving towards more generic robots with built-in intelligence and tactile feedback that can adapt to new tasks at hand, often based on data-driven methods such as reinforcement learning. This opens the door for robotic applications in many tasks within manufacturing industries such as retail warehousing, forestry, health care, service robotics, recycling waste management and space engineering. Or environments that are directly harmful for humans to perform such as dismantling bombs and demolition of old nuclear plants. To handle these tasks, robots need grippers that are versatile and more similar to our human dexterous hands. Grasping, gripping, lifting and in-hand manipulations of objects are fundamental movements that human hands can easily perform without careful planning and control. The more dexterous the task, the harder it is to perform for a robot which is why the task of grasping, gripping and dexterous manipulation is one of the most important challenges in the research field regarding versatile robots today.

A robot's ability to grasp and manipulate objects depends on the manipulator's geometric design, physical properties and controller, in relation to the object and the task. It is advantageous if most of the experimentation and development takes place

in a simulated environment, where design, motion planning, contact models, etc., can be pre-evaluated. For the results to be transferable to the physical domain and robot control using artificial intelligence, realistic and fast physics simulations are necessary. Algoryx's physics engine AGX Dynamics, AGX for short henceforth, is particularly well suited for industrial applications in robotics, mechatronics, and heavy vehicles - where articulated and motorized mechanisms interact in unstructured environments through dynamic contacts.

This thesis aims to benchmark the abilities of AGX regarding the physics needed in grasping and gripping by validations against real experiments, as well as building knowledge for the future to be able to achieve real-time dexterous manipulation in simulation with results that transfer to reality.

1.1 Research problem

Algoryx is one of the world-leading providers of software for visual and interactive physics-based simulation. Their physics engine AGX with its numerical integrator SPOOK [1] is used with great success in fields such as heavy vehicles, off-shore industry, training simulators, industry robots and material handling. Although, simulations of robotic grasping and gripping in AGX have not been sufficiently explored. Grasping and in-hand manipulation simulations are currently markets of growth as a result of the need for versatile robots with dexterous hands.

When simulating grasping, the users of AGX are challenged with choosing suitable geometric representations and calibrating stiffness, damping and friction as well as fine-tuning settings for the numerical solver to reach optimal performance. As of now, it is unknown if the difficulties are only connected to making suitable choices or if there are limitations in the implemented contact models, collision detection or numerical methods. Moreover, given the variety of solvers, friction and contact models available in AGX, it is hard to tell which one outperforms for a given task since it has not been well explored.

The key functionalities for successful robotic grasping simulations consist of robust collision detection, which is highly important for grasping tasks; constraint stability and simulation speed for systems with a high number of constraints, e.g. for joints; as well as the determination of friction forces and slip velocity with a realistic friction model. Grasping is a movement that contains complicated contact dynamics which need to be modelled both for simulation speed and accuracy. To accurately simulate systems with many quantities involved, it is important to get the foundations right. Friction is one of

those properties which arises from surface asperity and can be thought of as the sum of very tiny forces along the edge of a surface. The way we model physical phenomena is always an approximation in some way or form, and friction is no exception to that. To capture it well requires the use of empirical methods for analysis and development of theory. Preliminary studies and tests done when working on this thesis have shown that friction is a very important feature in robotic grasping and a challenging phenomenon to simulate perfectly accurately. Hence it will be one of the largest focus points in this work.

Learning from the existing challenges in AGX simulations, the research problems for this thesis are summarized as follows:

- How are the current friction models available in AGX performing on simple dynamics involving contacts and friction? Are there any significant limitations? If so, what kind of measures should be taken to overcome the limitations?
- In which ways can contact dynamics be modelled to best suit grasping tasks?
- Which benchmarks are suitable to validate rigid multibody dynamics simulations and how does AGX perform on these?

1.2 Purpose and goal

The goal of this thesis is to benchmark AGX's abilities to simulate important contact mechanics involved in robotic grasping. Based on the results of benchmark tests, the question of where AGX stands today in terms of strengths and opportunities for improvements should be answered.

Furthermore, the aim is to increase the knowledge about contact models for grasping tasks and find out possibilities for advances in this field.

1.3 Research approach

To address the research problems listed above, a series of benchmark tests are performed with AGX. The approach to answer the questions and research tools used are briefly described here.

Validation: Experimental data sets have been acquired from MCube Lab at MIT [2], where a series of in-hand manipulation tasks have been performed. The same setup is recreated in AGX for validation. The first benchmark is planar pushing, that is, pushing an object on a flat surface with an external force. In this case, a rod attached

to a robot arm. The second benchmark is prehensile linear pushing, where a gripper is holding an object in a grip and pushes it against the environment to achieve a different grip. A motion that is beneficial to use in a series of similar tasks to manipulate objects.

Modelling: The simulation scenes are built in AGX with Python scripts, where objects are given geometric representation and physical attributes such as contact material properties. Contact forces and object poses are extracted by a step event listener.

Analysis: To find out how well suitable AGX is for simulating grasping, the experimental results are compared with the position of the object and friction forces from simulation. Error analysis from the simulation results is performed.

Theoretical investigation: An in-depth understanding of the friction models from the theoretical perspective is gained via a thoughtful literature study. Such a theoretical investigation contributes to the new inspirations for analysis and improvement of the friction models implemented in AGX.

Testing new friction models: Based on the results from the validation of AGX against real-world experiments, possible improvements to the existing friction models are made. The performance of the improved models is investigated and presented.

Chapter 2

Robotic grasping

The task of grasping can be divided into a sequence of smaller tasks. For a robot to be able to manipulate an object autonomously it needs to be able to: perceive the object and locate it; find a collision-free path towards it; determine the best way to grip and make a stable grasp; possibly shift the object in the grasp; and finally plan a trajectory and move it to the desired location. This sequence can be simple if the objects are known and the manipulation task is exactly the same every time. However, if the object's physical properties are unknown to the robot and it needs to adapt and learn from its environment, sensors and data-driven methods are beneficial. To model a dexterous manipulation sequence analytically is complex. If the robot instead can learn the skill, like a human child growing up does, one can get around some of the challenges connected to the lack of accurate models. This requires a lot of training data and it is highly beneficial to first implement algorithms and run testing in a simulated environment before it is transferred to a real experiment [3].

The end effector of a robot arm is the device of which the robot interacts with its environment and could be a gripper or some kind of tool. How it is designed is determined by its applications but for grasping tasks, the jaw gripper is still the most commonly used. Without a functional hand, a robot is basically helpless so the development of complex robotic end effectors over the last decades is one of the reasons for the successful use of robots in industrial automation. Complex end effectors demand functional control sequences though in order to fully utilize their dexterity.

With inspiration drawn by the dexterous hands of us humans, robotic grippers are becoming more complex than the traditional jaw gripper and can now perform complicated tasks which can contribute to human lives in many ways. For example, in Figure 2.1a the robot hand, controlled by brain activity, can assist a paralyzed person

to drink [4], and thus improve quality of life. The automated industry robot, such as a forestry crane manipulator [5], can enhance the efficiency of material handling and lead to better economic benefits. The robot hand by Open AI can solve Rubik's cube while holding it in a grasp [6], a task that requires a high level of dexterity. Also, well-designed robot manipulators can complete unsafe and impossible tasks for humans, such as collecting space debris [7] as in Figure 2.1g.

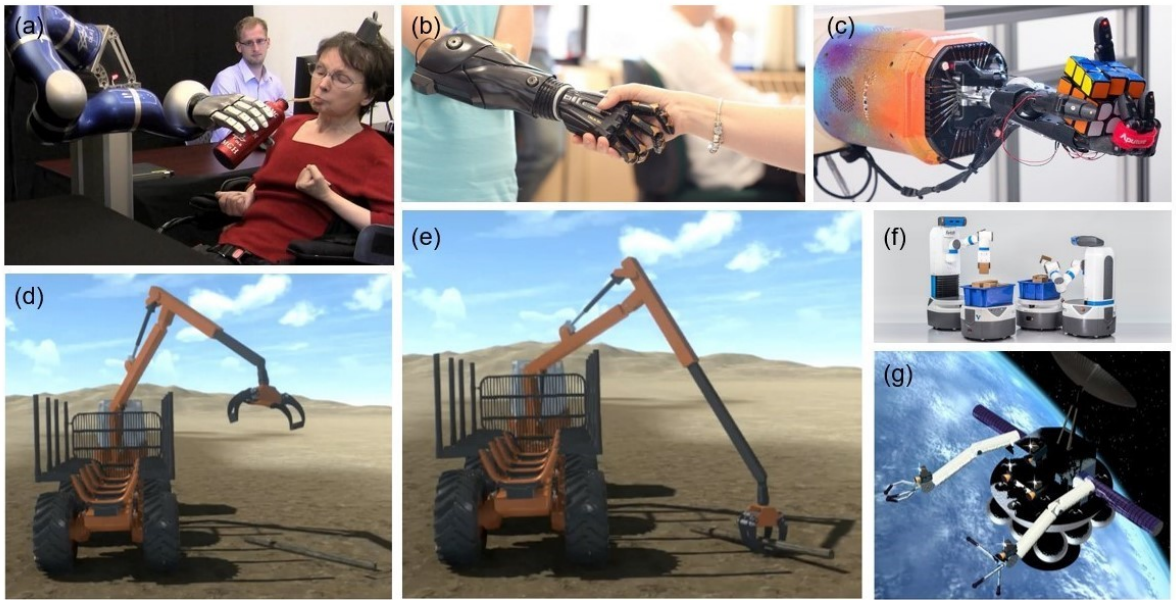


Figure 2.1: Examples of robotic grasping applications: (a) Assisting a paralyzed person; (b) Advanced prosthetics from Open Bionics; (c) Solving Rubik's cube; (d-e) Forest forwarder; (f) Fetch Robotics' robots assembling packages in a warehouse; (g) Collecting space debris.

2.1 Contact mechanics with friction

The dynamics and contact mechanics of grasping and manipulation are challenging to capture in simulation with high accuracy and performance. Simulations of grasping usually depend on rigid body mechanics, where both hand and the grasped object are modelled as rigid bodies. Forces transmitted through the contacting bodies include normal forces and tangential forces due to friction. The contact itself can in simulation be represented in a few different ways, with the point contact approach being the most common one because of its simplicity and computational speed [8]. The contact is specified by several contact points, which is a simplification since the initial point of contact turns into an area if the bodies stay connected and the bodies might deform depending on contact forces and material characteristics. This is further explained in Section 2.1.1. Two bodies are considered to be in contact when their relative distance

becomes zero. During and after the initial impact, friction must be considered, which can include a change from sticking to sliding or the other way around.

A grasping contact with soft fingertips differs from hard fingers since the contacting area increases as the normal force increases, which creates more traction and the possibility to apply a frictional moment about the contact normal [9]. The force along a contact normal should only be a repulsive one and always be zero if the objects are moving away from each other. Hence contacts can be treated as unilateral constraints with complementarity conditions which is elaborated in Chapter 3.

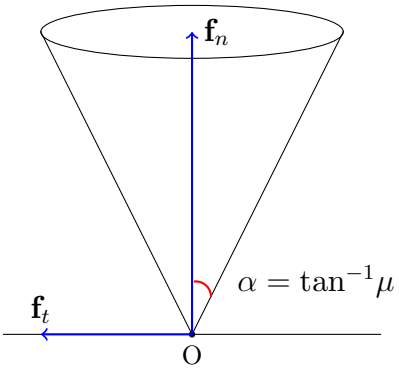
Friction is an important phenomenon to consider when studying grasping since frictional forces play a big role in a stable grip. It is also a feature that is challenging to capture analytically and model for simulations with good computational speed and accuracy. This is why frictional contacts are studied deeply in this thesis with a focus on grasping applications.

The Coulomb friction model for contacting surfaces is one of the most commonly used in simulation. It states that the frictional force cannot be larger than the normal force times the friction coefficient and hence set bounds to the magnitude of the friction force. The model is an empirical, approximate model of dry friction, which cannot fully describe the complicated mechanics of contact at a microscopic scale, but still performs well because of its simplicity and ability to capture most of the behaviour of surfaces in contact. More complex models have been developed by using the Coulomb formulation as a basis, and when to apply which model depends mainly on the intended application [10]. Geometrically, the set of all forces that can be transmitted through a contact with the Coulomb friction model, can be visualized with a friction cone like the one shown in Figure 2.2a. Accordingly, the friction coefficient, which is determined from the material characteristics of the surfaces in contact, is interpreted and visualized as the angle between the normal force and the edge of the friction cone.

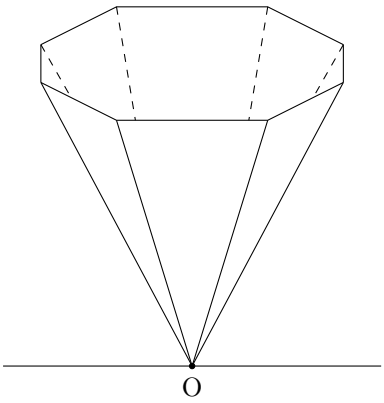
For computational purposes, the quadratic friction cone is usually simplified with a polyhedral, for instance a pyramid, such that the contact plane can be parameterized with a number of vectors. More cone edges provide a better approximation but increase computational time. More on this is presented in Section 3.3.

2.1.1 Point contact vs. area approach

When using a point contact model to calculate forces between contacting objects, discrete contact points are created at a number of points on the intersecting surface. The model then provides constraints on forces applied at a certain contact point and



(a) The smooth Coulomb friction cone for a point contact moving towards the right, where $\mathbf{f}_n$ is the normal contact force, $\mathbf{f}_t$ the friction force, μ the friction coefficient and O the point of contact.



(b) Linearized approximation of the friction cone with eight edges.

Figure 2.2: A geometrical representation of the Coulomb friction law.

can be more or less complex depending on if a frictionless contact, a frictional or a soft contact is considered. In the frictionless case, the constraint forces can only be exerted along the contact normal while a frictional contact also can have tangential forces in the contact plane. A soft point contact model also has torsional friction around the contact normal. The tangential and torsional contact forces are usually constrained by the Coulomb friction law. These forces need to be solved in simulation.

Area based contact models are often inspired by Hertz's contact theory developed during the 19th century. Since then many compliant models have been introduced with various results of approximating human fingers and how they deform upon contact [11]. Danielczuk et al. [8] benchmarked three different contact models and compared them to their own proposed area based model in terms of stable grasp estimations. The results show that a point contact model is much faster than an area based. However, the point contact model leads to more false negatives when predicting if a stable grasp can be performed. That is, the model estimates failure when in fact it should be possible to grasp the object since it does not consider the full range of contact wrenches.

2.1.2 The effect of geometry

The geometry of a manipulated object can make a significant difference in how the final pose turns out after a grasping procedure. In the work by Ciocarlie et al. [9]

two identical manipulation procedures, where a robot hand is gripping an object on a table with two fingers and lets it slip into the palm in a grasp, turns out entirely different depending on the local curvature at the contact points. The hand is grasping a rectangular object which does not fall into the palm because the friction forces prevent rotational slip. When it is grasping a cylindrical object, however, the contact area between the hand and the object is much smaller and restricts the maximum possible magnitude of the frictional torque. The friction force can in that situation not withstand the rotation and the object falls into the palm of the robot's hand.

Grasping of objects with curvature could benefit from deformable gripper jaws. However, this results in nonplanar surface contacts and a six dimensional frictional wrench instead of the three dimensional used for planar contacts. A model for nonplanar frictional contacts is e.g. presented by Xu et al. [12].

2.2 Robotic grasping simulations

In the review by Billard et al. [13] of trends in robot manipulation, it states that there is still a need for basic theoretical development to achieve dexterous manipulation with robotic hands. For instance, they call for better modelling of soft point and area contacts, and contacts for soft and deformable objects, something they conclude existing simulators are not yet able to deliver. Data-driven methods need to be better at simulating soft bodies and provide models for objects whose state changes after manipulation, e.g. plastic deformation. There is still a long way to go for robots to be able to fold clothes or chop vegetables.

In simulation, rigid object models are the most common because of their computational speed compared to models with elasticity and bulk deformation. Models based on rigid point contacts are fast but non-smooth, which for instance makes them hard to apply to objects with arbitrary geometries. Instead, there are different approaches of locally violating the rigid body assumptions and calculate for contact forces by having: point contacts with collisional restitution and friction; discrete spring-dampers, using the Hertz contact law; or finite-element models for contact regions [14]. Point contacts obviously miss the area-dependent physical phenomenon that occurs between contacting bodies so a fast and stable model that accurately includes this would be very beneficial for robotic applications especially grasping. Elandt et al. [14] propose the use of a pressure field model to capture area-dependent phenomena by combining ideas from an elastic foundation model, using springs, and hydrostatic pressure. Rigid objects are given an interioral pressure field which is equal to a contacting object's

field at the contact surface when overlapping. Tractions at the contact surface are then determined with a dissipative rate-dependent pressure and friction, and contact wrench with an integral over those tractions. They conclude that their model is sufficiently fast, robust and physically accurate for use in robotics applications. However, the target of their model is to behave like a physical system but not necessarily predict any particular experiment perfectly.

An augmentation of the classic Coulomb model for point contacts is presented by Bouchard et al. [15] where the intersecting volume of two rigid bodies are used to compute a contact normal based on the volume gradient. It results in a six degrees of freedom constraint that provides both translational and rotational torque due to dry friction. When two objects collide there is usually a slight overlap of their geometries and this overlapping volume is what is used to constrain rolling and spinning torque. Together with a point contact Coulomb model it allows for more realistic frictional behaviour with a single six dimensional contact representation. It is shown to eliminate redundant contacts which can occur with a point contact model and by that reduce the size of the system of equations.

Open AI [6] demonstrated that a real human-like robot hand can be trained only in simulation to perform a manipulation problem of high complexity: solving the Rubik's cube. This was done with reinforcement learning and what they call automatic domain randomization (ADR). The algorithm for ADR randomizes certain aspects of the environment such as the visual appearance of the Rubik's cube or the dynamics of the robot's hand with the motivation that training on diverse distributions over environments leads to better simulation to reality (sim2real) transfer.

Piazza et al. [16] suggest that a paradigm shift in manipulation from a rigid to what they call a 'soft' approach is about to take place. In rigid manipulation, the robotic fingers are brought to the object and generates a grasp without interacting with the environment. Soft manipulation exploits the environment and its enabling constraints to create a more complex interaction with the objects. Chavan-Dafle et al. [17] also explore this experimentally with push-primitives that use the environment to help with a manipulation task. This is called prehensile pushing and can be utilized to improve the abilities of robotic in-hand manipulations. Kolbert et al. [18] have extended on that work and evaluated contact models and their abilities to predict motion and forces involved in robotic manipulations with focus on simple in-hand actions. One of these, the linear pushing, is also used as a benchmark to validate AGX in this thesis.

Chapter 3

Computational methods and models

AGX Dynamics is a simulation library centred around rigid multibody dynamics derived from discrete variational Lagrangian mechanics. Rigid body dynamics are in classical mechanics described by the Newton-Euler equations of motion. These are ordinary differential equations (ODEs) which can become stiff if coupling bodies are modelled in terms of explicit force models. In AGX joints between bodies are modelled with kinematic constraints. A kinematic constraint decreases the degrees of freedom and describes how the relative motion between jointed bodies in the system is restricted. To each constraint, a constraint force is associated, which does not remove or add energy to the system but exists to keep the given constraint satisfied.

3.1 Rigid multibody dynamics

Let $\mathbf{x}$, $\mathbf{v}$ and $\boldsymbol{\lambda}$ be the state variables for a rigid multibody system where $\mathbf{x} \in \mathbb{R}^{n_d}$ is a coordinate vector with time derivative $\dot{\mathbf{x}} = \mathbf{v}$ and $\boldsymbol{\lambda} \in \mathbb{R}^{n_d}$ a Lagrange multiplier associated with a constraint. The equations of motions are then given by the differential algebraic equations (DAEs)

$$\begin{aligned} \mathbf{M}\dot{\mathbf{v}} &= \mathbf{f}_{ext} + \mathbf{G}^T\boldsymbol{\lambda}, \\ \mathbf{g}(\mathbf{x}, \mathbf{v}) &= \mathbf{0}, \end{aligned} \tag{3.1}$$

where $\mathbf{M} \in \mathbb{R}^{n_d \times n_d}$ is the mass matrix of the system with n_d degrees of freedom, $\mathbf{f}_{ext} \in \mathbb{R}^{n_d}$ external forces on the bodies, $\mathbf{g} \in \mathbb{R}^{n_d}$ is the constraint function and $\mathbf{G} = \frac{\partial \mathbf{g}}{\partial \mathbf{x}}$ the constraint Jacobian. The constraint force $\mathbf{G}^T\boldsymbol{\lambda}$ inhibits motion orthogonal to the constraint surface $\mathbf{g}(\mathbf{x}) = \mathbf{0}$, where the Jacobian determines the force's direction while the Lagrange multiplier determines its magnitude. Valid velocity and acceleration is

$\mathbf{Gv} = \mathbf{0}$ and $\dot{\mathbf{G}}\mathbf{v} + \mathbf{G}\dot{\mathbf{v}} = \mathbf{0}$, where $\dot{\mathbf{G}} = \partial_x \mathbf{G} + \partial_t \mathbf{G}$.

Equation (3.1) in matrix form becomes

$$\begin{bmatrix} \mathbf{M} & \mathbf{G}^T \\ \mathbf{G} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{v}} \\ -\boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{f}_{ext} \\ -\dot{\mathbf{G}}\mathbf{v} \end{bmatrix}, \tag{3.2}$$

where the zero on the main diagonal causes the matrix to be ill-conditioned in many situations. To counter this a small perturbation is introduced, replacing the zero by a regularization $\boldsymbol{\Sigma}$, which causes the system to be slightly compliant but ensures numerical stability.

3.1.1 Contacts

Contacts between bodies are treated as unilateral constraints where penetration is not allowed. A normal contact constraint is defined as

$$g_n \geq 0, \quad \lambda_n \geq 0, \quad g_n \lambda_n = 0, \tag{3.3}$$

where $g(\mathbf{x})$ represents the distance between bodies and λ_n the normal interaction force associated with the contact. An equivalent formulation to Equation (3.3) using complementarity is

$$0 \leq g_n \perp \lambda_n \geq 0, \tag{3.4}$$

where $\perp$ here is the complementarity operator. In complementarity theory, only one of the factors can exist at each time step which means that if we have an acting contact force, $\lambda_n > 0$, the contact constraint function must be fulfilled i.e. $g(\mathbf{x}) = 0$ and vice versa. This mathematical framework can represent that contact forces between bodies only exists upon contact and must vanish when they separate. It can also model transitions between sticking and sliding contacts.

3.1.2 Regularization, stabilization and time stepping

To introduce a regularization like $\boldsymbol{\Sigma}$ to the system in Section 3.1 makes it not only easier to solve but can also be used to model elasticity of contacts. Each constraint is replaced by a slightly perturbed one with the compliance parameter ϵ as $\mathbf{g}(\mathbf{x}) = \mathbf{0} \rightarrow \mathbf{g}(\mathbf{x}) = -\epsilon\boldsymbol{\lambda}$. Still, numerical errors might cause constraint drift and to keep motion on the constraint surface a stabilization term can be added to the constraint. This is conventionally treated with a Baumgarte stabilization which modifies the constraint to $\ddot{\mathbf{g}} + 2\alpha\dot{\mathbf{g}} + \beta^2\mathbf{g} = 0$. Although these extra terms restore

and damp motion orthogonal to the constraint surface, difficulties remain in finding appropriate values for the parameters α and β. Tuning these in a bad way can tamper both with the numerical stability and cause modelling errors which alters the physics.

The numerical integrator used in AGX is called SPOOK [1] and is a discrete time variational stepper of Verlet type derived from the least action principle. The method combines constraint regularization motivated by physics and stabilization terms to achieve both stability and physical elasticity. Here the constraints are introduced as massless particles, called spooks, with potential energy

$$U(\mathbf{x}) = \frac{1}{2}\mathbf{g}(\mathbf{x})^T \epsilon^{-1} \mathbf{g}(\mathbf{x}), \tag{3.5}$$

with ϵ again as a compliance parameter which works as inverse stiffness to the potential. An increased value of ϵ makes the constraints more compliant and can be connected to elasticity in contact constraints. Note that the gradient of the potential gives back the constraint force

$$-\frac{\partial U}{\partial \mathbf{x}} = -\mathbf{G}^T \epsilon^{-1} \mathbf{g}(\mathbf{x}) = \mathbf{G}^T \boldsymbol{\lambda}. \tag{3.6}$$

For regularization, a Rayleigh dissipation function $R = \frac{1}{2}(\mathbf{G}\mathbf{v})^T \tau^{-1} \mathbf{G}\mathbf{v}$ with damping parameter τ is introduced. With SPOOK and linearization of the constraints we arrive at the mixed linear complementarity problem (MLCP)

$$\begin{aligned} \mathbf{H}\mathbf{z} + \mathbf{b} &= \mathbf{w}_l - \mathbf{w}_u \\ 0 \leq \mathbf{z} - \mathbf{l} &\perp \mathbf{w}_l \geq 0 \\ 0 \leq \mathbf{u} - \mathbf{z} &\perp \mathbf{w}_u \geq 0, \end{aligned} \tag{3.7}$$

where $\mathbf{l}$ and $\mathbf{u}$ are upper and lower limits allowed for $\mathbf{z}$, which is to be solved for. The slack variables $\mathbf{w}_l$ and $\mathbf{w}_u$ are used internally by the MLCP solver and originates from the mathematical optimization theory of linear complementarity problems (LCPs). The other matrices in Equation (3.7), in discretized time with k denoting the current time step, are

$$\mathbf{H} = \begin{bmatrix} \mathbf{M} & -\mathbf{G}_t^T & -\mathbf{G}_n^T \\ \mathbf{G}_t & \boldsymbol{\Gamma} & 0 \\ \mathbf{G}_n & 0 & \boldsymbol{\Sigma} \end{bmatrix}, \quad \mathbf{z} = \begin{bmatrix} \mathbf{v}^{k+1} \\ \boldsymbol{\lambda}_t^{k+1} \\ \boldsymbol{\lambda}_n^{k+1} \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} -\mathbf{M}\mathbf{v}^k - h\mathbf{f}^k \\ 0 \\ \frac{4\gamma}{h}\mathbf{g}^k - \gamma \mathbf{G}_n \mathbf{v}^k \end{bmatrix}, \tag{3.8}$$

where the subscripts t and n represents tangential and normal directions and h the time step size. Constraint regularization and stabilization $\boldsymbol{\Gamma}$, $\boldsymbol{\Sigma}$ and γ introduces elasticity

and damping to the system and is represented as

$$\mathbf{\Gamma} = \frac{1}{h}\text{diag}(\gamma), \quad \mathbf{\Sigma} = \frac{4}{h^2}\text{diag}[\epsilon/(1+\frac{4\tau}{h})], \quad \gamma = \frac{1}{1+\frac{4\tau}{h}}. \tag{3.9}$$

It is possible to connect the constraints to follow the Hertz contact law and map the regularization parameters to physical properties e.g. Young's modulus of a material.

3.2 AGX multibody dynamics

Rigid multibody dynamics simulations in AGX are centred around the rigid body. To the rigid body, a geometry is assigned which contains a shape e.g. box, cylinder or sphere. The geometry is the entity that can collide with other geometries and create contacts. The contact mechanics are governed by a set of local contact constraints, which are created when two geometries overlap: a contact is detected. This approach will cause a slight interpenetration between contacting bodies, which models local linear deformation. The depth is determined by external forces and material parameters such as Young's modulus.

A number of contact points between the colliding geometries are created upon which the contact forces are applied. For a simple simulation of a box resting on a plane, AGX will create four contact points between the two objects, one at each corner of the box. These points are a discretization of the contact area. The contact forces at one of these contact points consist of a normal force and two tangential forces. This point contact approach is computationally fast and stable for objects larger than around 1 dm. When dealing with smaller and lighter objects the area based approach can be better suited. A contact point can be considered as a spring with a contact stiffness per unit area defined as

$$k = \frac{E}{h}, \tag{3.10}$$

where E is the Young's modulus of the contact material and h the material rest length, the total length of the bodies in the contact direction. With this and a contact depth, d, given that there is an overlap, the local pressure is calculated as $P = kd$. And finally the contact force as

$$F = PA, \tag{3.11}$$

where A is the contact area. This elastic foundation model used in AGX is a simplification of the one presented by Perez et al. [19] and turns into the point contact model by setting the value of h to 1.

3.2.1 Friction models in AGX

The user can choose from a few different friction models to determine how the tangential forces in the contacting plane are solved. Note that AGX uses the same friction coefficient for both static and dynamic friction.

Box friction is one of the simplest approximations of the Coulomb friction law used in simulation. The mathematics of it is presented in Section 3.3. The tangential forces are given box bounds, $-\mu\,\mathbf{f}_n \leq \mathbf{f}_t \leq \mu\,\mathbf{f}_n$, which are calculated based on the normal force, $\mathbf{f}_n$, and friction coefficient μ. The nominal case is when the contact was created at an earlier time and the normal force from the previous time step is used to set the bounds of the friction box. Otherwise, AGX estimates the normal force by the relative impact speed if a contact is new. These bounds are then static throughout the solve stage. For a box on a plane where four contact points are created, one at each corner of the box, the box friction model creates bounds at each contact point as illustrated in Figure 3.1. The orientation of the boxes is defined by a primary and a secondary friction direction, which are orthogonal to each other. Unit vectors in these directions span the contact plane.

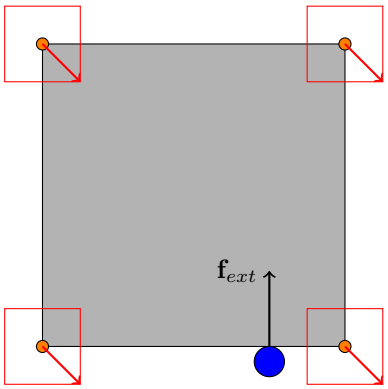


Figure 3.1: Box friction in action for a box being pushed on a plane by an external force. The red boxes at each corner contact point represents the box bounds of the reaction friction forces arising from the action. Since the force is applied off-center, the box would like to rotate and maximum friction force is achieved if they direct themselves towards the corners of the boxes.

In the **Scaled box friction** model the box bounds can be scaled during the solver iterations. The current normal force i.e. the correct one is achieved as a callback from the solver to set new bounds. This method is computationally more expensive but in return more realistic.

The final model is the **Iterative projected cone friction** model. This is used as default in AGX but can only be utilized if an iterative solver type is used. Normal forces are solved by the direct solver and then tangential friction is solved iteratively.

If a contact is slipping this method projects friction forces onto the friction cone such that you always get friction force = friction coefficient × normal force, i.e. the limit of the Coulomb friction law.

Sundling [20] have previously shown that both box friction models in AGX have anisotropic behaviour in the contact plane. The iterative cone model does not have this limitation but behaves isotropically.

3.3 Friction

Friction is introduced as a velocity constraint on the rigid multibody system in the tangent space to the contact normal as

$$\mathbf{G}_t\mathbf{v} = 0. \tag{3.12}$$

The Coulomb friction law is commonly used to model friction when simulating dynamics of rigid and flexible bodies. It is stated as

$$\|\mathbf{f}_t\| \leq \mu\|\mathbf{f}_n\|, \tag{3.13}$$

where $\mathbf{f}_t$ is the friction force, μ the friction coefficient and $\mathbf{f}_n$ the normal force. The law establishes the maximum possible friction force at a contact point depending on the normal force of that contact. Geometrically you can describe this as a friction cone. Imagine a contact plane between two bodies and let $\hat{\mathbf{n}}$ be the normal to that plane and $\hat{\mathbf{u}}$ and $\hat{\mathbf{v}}$ be the tangential basis vectors of the contact plane. Then at a contact point on the plane let $\mathbf{f} = [f_n, f_u, f_v]$ be the net contact force and $\boldsymbol{\nu} = [\nu_n, \nu_u, \nu_v]$ the relative velocity of the contact. The friction cone (FC) at a point contact is then defined as the following set of vectors:

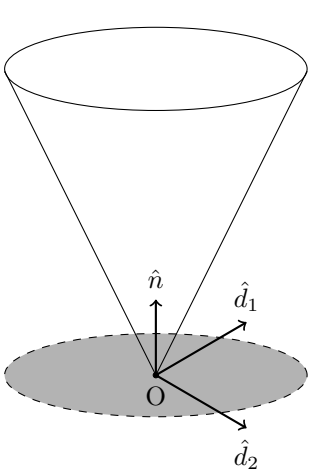
$$\text{FC} = \{f_n\hat{\mathbf{n}} + f_u\hat{\mathbf{u}} + f_v\hat{\mathbf{v}} \,|\, f_n \geq 0,\ f_u^2 + f_v^2 \leq \mu f_n^2\}. \tag{3.14}$$

The Coulomb friction law is satisfied if the friction force is within the bounds of the friction cone. If a contact is sliding the contact force is on the boundary of the cone and the last inequality in Equation (3.14) is strict. For anisotropic friction, the cone turns into an elliptic cone. As we can see Equation (3.14) also turns the dynamics of a system subject to Coulomb cone friction to be governed by a quadratic constraint and becomes a nonlinear complementarity problem (NCP). An NCP does not always have an existing solution and can be hard to solve, which is why the friction cone is usually linearized with some kind of approximation. One way is to replace the friction cone

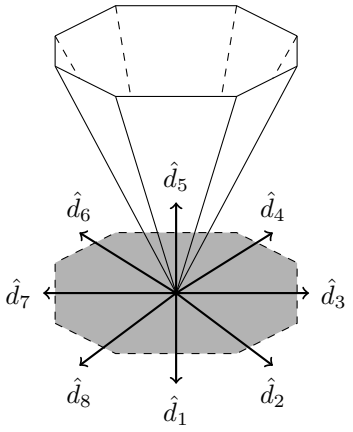
with a polyhedral approximation, which instead turns the equations into an LCP with many existing tools to solve. The set of vectors in this polyhedral cone are

$$\mathrm{FC}_p = \{f_n\hat{\mathbf{n}} + \mathbf{D}\boldsymbol{\beta} \,|\, f_n \geq 0,\, \boldsymbol{\beta} \geq 0,\, \mathbf{e}^T\boldsymbol{\beta} \leq \mu f_n\}, \tag{3.15}$$

where $\mathbf{D}$ contains the direction vectors $\mathbf{d}_i$ which spans the friction plane as shown in Figure 3.2, $\boldsymbol{\beta}$ is used as a weighting vector for the directions and $\mathbf{e} = [1, 1, \ldots, 1]^T \in \mathbb{R}^k$, where k is the number of edges in the polyhedral approximation of the friction cone [21].



(a) The smooth friction cone.



(b) Linearized approximation of the friction cone with eight directions.

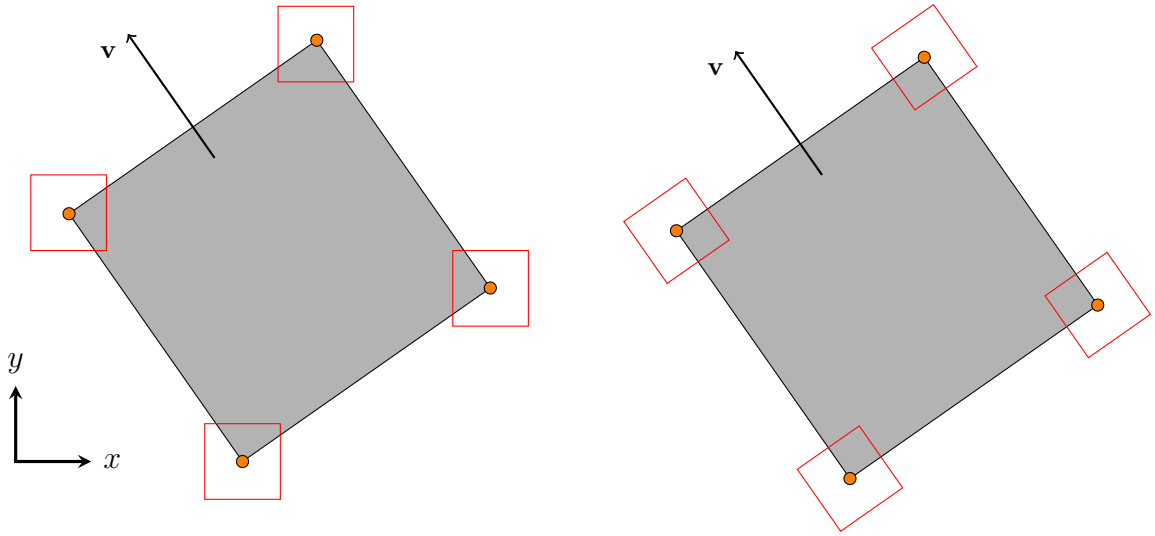
Figure 3.2: Geometrical representations of the Coulomb friction law.

3.3.1 Oriented friction models

When using the box friction model the friction box is always oriented with the global axes of the simulation at each contact point. To get around this, it is also possible to define the orientation of the friction box in any way the user likes e.g. to have it co-moving with a body. The orientation is set by defining the primary friction direction and then the secondary direction is created to be orthogonal to that. This relative orientation stays for the entire simulation. An illustration of this is given in Figure 3.3 where both standard orientation and user-defined is found.

3.4 Solvers

For each contact, different solve types can be defined by the user of AGX. That is, constraints such as friction and normal equations can be solved in many ways. The **Direct solver** solves the system of equations to machine precision, with the exception



(a) Standard friction box alignments.

(b) Friction boxes aligned with the movement.

Figure 3.3: The the standard box friction model vs. the oriented box friction.

of rounding errors. It is a block-sparse solver with LDLT fill-reducing factorization and direct pivoting. It is a very stable solver with high precision even for systems with large mass ratios, that is, systems that include objects with masses that differ in many orders of magnitude.

The **Iterative solver** uses the projected Gauss-Seidel (PGS) method, which is an extension of Gauss-Seidel that is used to solve linear equations. It solves the equations one by one, where one iteration is one pass at all equations involved and has a very high speed. The algorithm proceeds until a good enough solution is found or a maximum number of iterations are reached. It is scalable for large systems but has a poor convergence rate and leaves a residual that causes artificial elasticity and frictional sliding errors. Also, it might encounter problems for systems with large mass ratios or systems that have high stiffness which further increases the rate of convergence.

The **Split solver**, which is the default solver in AGX, is the best of two worlds. It utilizes the direct solver on joints and normal equations while solving for friction iteratively.

3.5 Improved friction implementations

The Box friction models give rise to anisotropy in the tangent plane, which can cause many issues in simulations. To avoid this, two new ideas of models which are presented here have been implemented and tested in AGX.

3.5.1 Randomized primary tangent direction

The randomized tangent model inherits from Oriented scale box friction and is a progression of that. Instead of giving the user the responsibility to choose a suitable primary friction direction, which stays for the entire simulation, the tangent plane is built randomly at each time step. The primary friction direction is rotated by a random angle $\theta \in [0, 360]$ from its default orientation and given a new direction at every step of the simulation. Over time this makes the friction box look more like a circle, like the one illustrated in Figure 3.4, and might remove some of the anisotropy in the tangent plane.

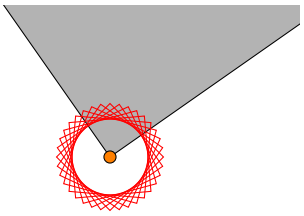


Figure 3.4: The Randomized tangent friction model over several time steps illustrated for one contact point.

3.5.2 Multi-directional tangent plane

Instead of having two possible friction directions, have as many as you want! The idea is to avoid the box and again create a more circular-like friction approximation. The user specifies how many directions they want and during the simulation, the implementation chooses the most suitable ones out of that set. In the first iteration when there is no friction, a primary direction is set and then the secondary direction is chosen from the number of other tangents created as the one most orthogonal to the first. When friction forces are in action, their direction is calculated and compared to the possible tangent directions in the set. The one most parallel then becomes the new primary friction direction and once again the secondary is chosen to be the most orthogonal one. Tests with the standard Oriented box friction model on the planar pushing tasks have shown that orienting the friction box with the slip velocity is better than orienting it with the principal axes.

Chapter 4

Benchmarks

Benchmarking is an important task to verify that your solver reflects reality when doing physics simulations and to validate the physics model. In this chapter, results from two validations against real data are presented and analyzed. Even simple push primitives like linear sliding, pivoting and rolling can expose issues and limitations in state-of-the-art contact modelling which is why the tests performed here are quite simple.

The first test performed is planar pushing, where an object resting on a flat surface is being pushed with a rod attached to a robot arm. The raw experimental data of this movement is provided by MCube Lab at MIT and contains thousands of pushes with varying object shape, surface material, speed, acceleration, contact position and push direction [22]. To minimize possible errors I have chosen to simulate the simplest setup: one rectangular object that is being pushed at a constant velocity of 10 mm/s and on one side only, but with different contact points on that side and push directions which will force the object to rotate.

The second test is involving the environment in a grasping task where the grip of an object is to be adjusted. This is called prehensile pushing and the experimental results are taken from the paper by Kolbert et al. [18].

4.1 Experimental procedure of planar pushing

The real-life pushing experiment and data collection performed at MCube Lab is described in this paper by Yu et al. [2]. To control the pushing interaction an ABB IRB 120 industrial robotic arm with 6 DOF was used. A stiff cylindrical steel rod is attached to the arm with length 156 mm and diameter 9.5 mm. This rod is further on called the pusher. In this validation process one of the eleven objects used in the

MCube Lab test will be pushed, a rectangle with dimensions $90 \times 90 \times 13$ mm and weight 837 g. A push is always 5 cm long and goes in a straight line. There are no curved push trajectories.

In the experiment, the dynamic friction coefficient between the object and the surface is determined by a series of tests. The theoretical dynamic coefficient of friction (DCoF) is defined as the ratio between the measured reaction force and the supporting normal force. For the rectangle that I have chosen, the value of DCoF was measured to be between 0.13 and 0.15 on a surface of ABS plastic. The value changed due to the surface being polished after some pushes. The coefficient between the pusher and the object was found in a variable slope experiment to be approximately 0.25. Both these values of the friction coefficient are used in AGX.

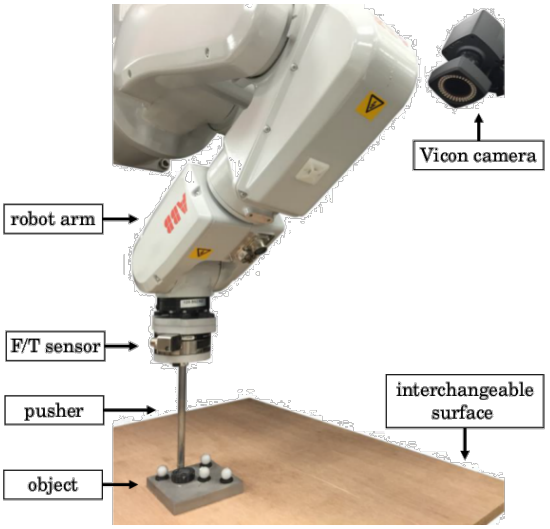


Figure 4.1: The experimental setup of planar pushing. Image source: MCube Lab website [22].

The data files from MCube Lab contain some notations to specify which type of push has been done, which is kept here for simplicity. The important ones is how push direction and contact point are specified. As seen in Figure 4.2b each side of the pushed object is divided into contact points between 0 and 1, where 0.5 corresponds to the middle of that side. The push direction is specified using the push angle notation, wherein the experiments the direction of the push has been varied between $-80°$ and $80°$ around the contact normal in increments of $20°$. Hence a straight push corresponds to this value being 0. The arrows in Figure 4.2c illustrate the direction of an angled push. Position of the centre of the object and force exerted on the pusher are provided given in x- and y-coordinates with respect to the coordinate system presented in Figure 4.2a.

4.2 Simulating planar pushing

The simulated push environment in AGX consists of three rigid bodies: a cylinder used as the pusher, which is a kinematic body controlled by giving it a velocity; an object that is being pushed, which is a dynamic body affected by the pusher; and a static ground that acts as the surface. Every push is then initiated with the same conditions

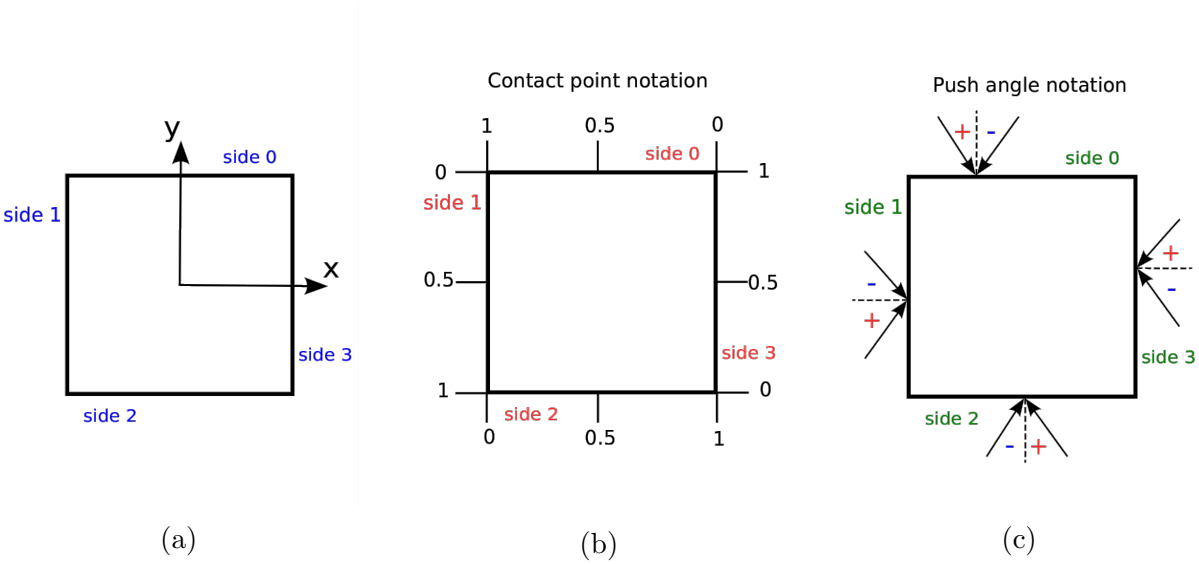


Figure 4.2: Defining the notations used for object's sides, contact points and push direction. Image source: MCube Lab website [22].

as in the experimental procedure regarding contact point and push direction. Velocity, acceleration and which side the push is applied to (side 1 is used) are held constant. The pusher's speed is always 10 mm/s and a push 5 cm long, which means all pushes have a duration of 5 s.

Four different pushes have been tested with a spread in contact points and push directions to possibly identify some issues. They are presented in Table 4.1 below.

Contact point	Push direction
0.5	straight
0.5	40°
0.8	straight
0.8	−20°

Table 4.1: The choice of pushes that have been tested.

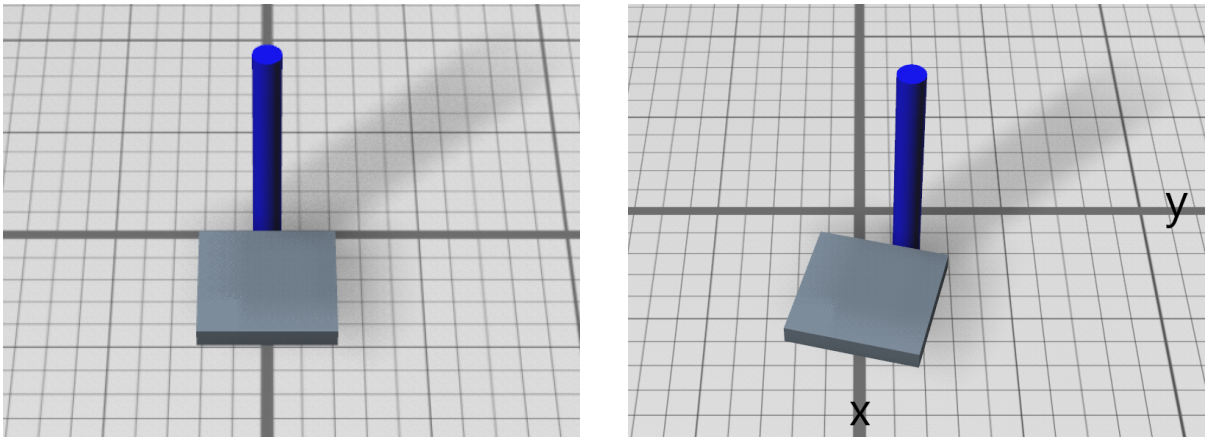
In order to specify contact characteristics between objects in AGX, a contact material is created. The contact material class contains information about material attributes of the objects in contact such as friction coefficient and compliance. A contact material is also used to define which solver type to use in order to solve for friction and also specify the friction model. Four contact points are created in the contact plane between the object and the ground, one at each corner of the box-shaped plane as was illustrated in Figure 3.1. Between the pusher and the object, two contact points are created. This contact proved to cause some trouble when only using the default settings in AGX, it

did not want to stay in continuous contact during the push. To resolve that the area based approach was activated and Young's modulus changed from its default value of 10^8 to 10^6, which caused a slight overlap but forced continuous contact between the objects.

4.2.1 Recording data

To compare results between the experimental pushes and the simulation, the position of centre of geometry and friction forces at each contact point are recorded during the push. Since there are four points where friction forces are created between the object and the ground, they are summed together into a resultant friction vector which then is divided into x- and y-components for comparison with the MCube Lab data. The reaction force between the pusher and the object should have the same magnitude and measurements showed that they differed from the surface contact with around 1 %. Since the contact was slightly tampered with to keep continuous contact the surface contact points were chosen to be the reference.

A push is controlled by giving the pusher a velocity, that determines speed and direction, which goes to zero when five seconds have passed. In Figure 4.3 initial and final position of a push with contact point 0.5 and a push direction of 40° is demonstrated. The pusher's initial position is at the origin of the global coordinate system.



(a) Initial position. (b) Final position.

Figure 4.3: Illustration of initial and final position of a simulated push shown in AGX Viewer for a push with contact point 0.5 and 40° push direction from the contact normal.

4.2.2 Solver and friction models

The default solver settings in AGX is to use the hybrid 'Split' solver with a time step of 1/60 s and the Iterative projected cone friction model. Only the solver type has been changed during the simulation tests in order to easier draw conclusions on occurring errors. Time steps down to 1/600 s have been tested on the direct solver to see if that is a parameter that could change the solutions significantly. However, it did not show any difference in the results so it is kept at default value to maintain real-time simulation.

The four different solver types (Direct & iterative, Iterative, Direct and Split) are tested on each push. If direct solves are used, the friction model uses the Scale box friction model during the direct solve and possibly iterative projection after the final direct solve. For the Iterative solver, 16 iterations are used, which is the default value.

4.2.3 Errors

Besides displacement and friction forces, AGX's ability to create a physical solution has been analyzed. Two errors are measured during the pushes: Coulomb cone error and alignment error, calculated in the same manner as in [23]. The Coulomb cone error is defined as

$$C = \max\left(0, \frac{\|\mathbf{f}_t\|}{\mu\|\mathbf{f}_n\|} - 1\right), \tag{4.1}$$

where $\mathbf{f}_t$ and $\mathbf{f}_n$ are the tangential and normal components of the contact force. If $C \geq 0$ it means that the friction force is correctly projected onto the friction cone. The object's alignment should be such that the net friction force and the sliding velocity vector, $\mathbf{v}$, are anti-parallel. The angle between those vectors are measured and the alignment error is then defined as

$$A = \arccos\left(\frac{\mathbf{f}_t \cdot \mathbf{v}}{\|\mathbf{f}_t\|\|\mathbf{v}\|}\right) - \pi, \tag{4.2}$$

where $A = 0$ means perfect alignment.

4.3 Planar pushing results

The four different pushes that have been done are compared with the MCube Lab data in terms of displacement and friction force in x- and y-direction. The results are presented per type of push in the following sections.

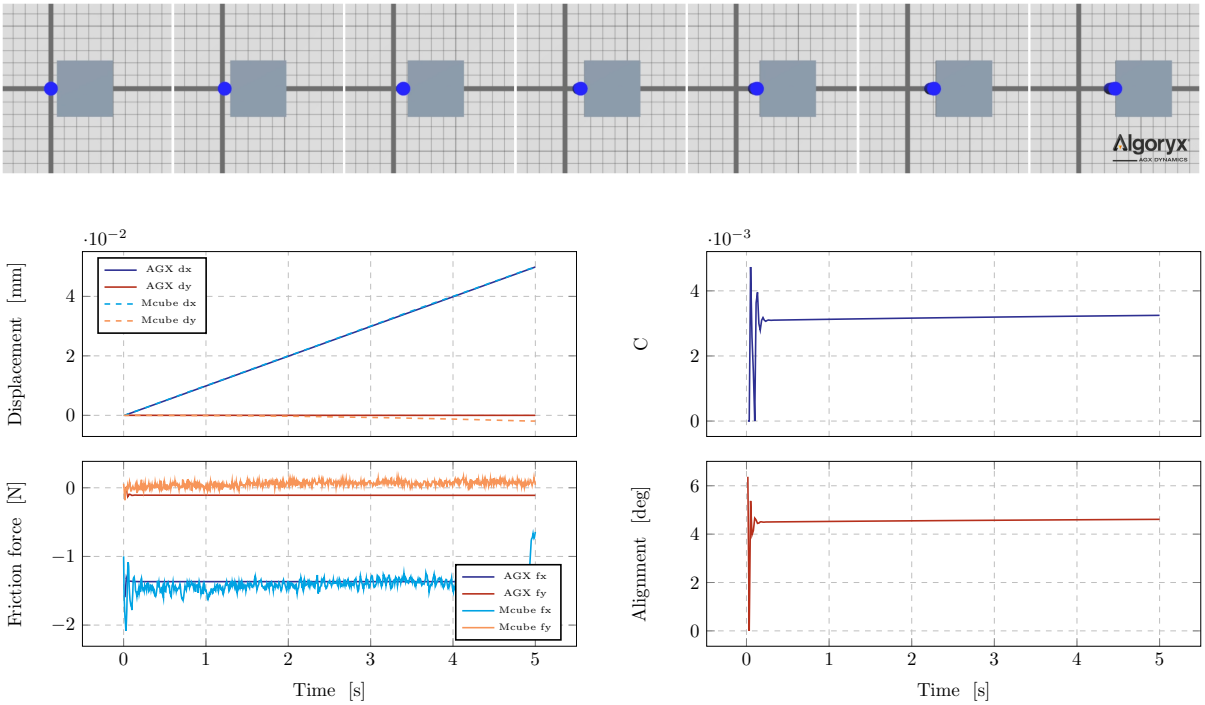
4.3.1 The trivial case

The first push that has been tested is a very straightforward one. The contact point is located right at the middle of the side and the push direction coincides with the x-axis. The top panel in all figures on the left-hand side shows the displacement in x and y for the simulated push in AGX together with the data from MCube Lab. For instance, we can see in Figure 4.4 that for this straight push the object moves almost only in the positive x-direction for 5 cm, which is to be expected. The other three solvers show almost identical results and are found in Appendix A.1. The MCube Lab data shows that the object starts to rotate ever so slightly at the end, which most likely is due to the real surface not being fully isotropic.

The bottom panel to the left in all figures presents the friction force divided into x- and y-components for the sum of all contact points. On the right-hand side, the two different errors are presented.

This push with the split solver, which is the default solver in AGX, has been used to calibrate the value of the friction coefficient set in AGX for the contact material between the object and the ground. To match the magnitude from the experimental force data the friction coefficient was set to be 0.16 during all pushes. The friction coefficient for the other contact material, between the pusher and the object, was set to 0.25, which is the same value as was found experimentally. Only a very small or very large value of that coefficient was found to affect the dynamics in a significant way, without changing any other parameter.

The image sequence on top of every figure is illustrating how the push looks from above in simulation.



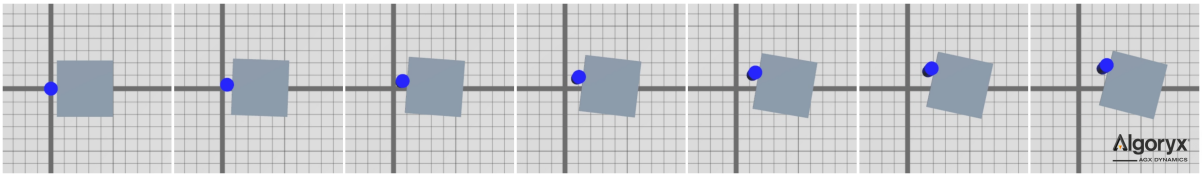
(a) Displacement and friction forces.

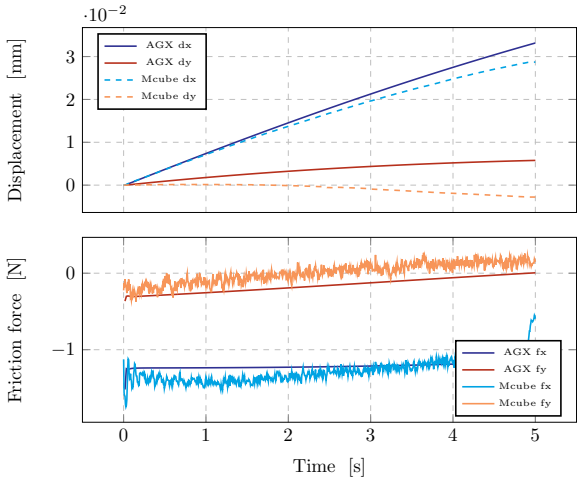
(b) Coulomb cone and alignment error.

Figure 4.4: Direct solver on a straight push with contact at the middle of the side.

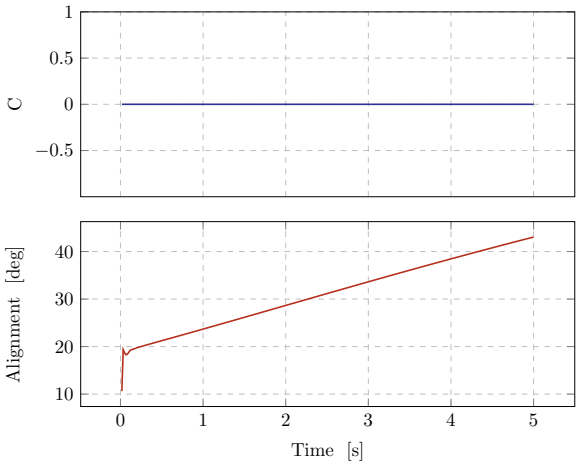
4.3.2 Angled pushes

Two of the pushes have a non-zero push angle with respect to the contact normal. When pushing an object at centre of mass with a push direction that deviates from completely straight forward, it should start to rotate and that is what is observed experimentally. Figure 4.5 and 4.6 illustrates the results of this push with the Split and Direct & iterative solver.



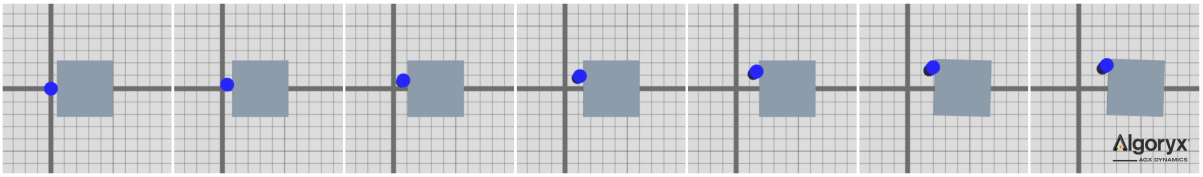


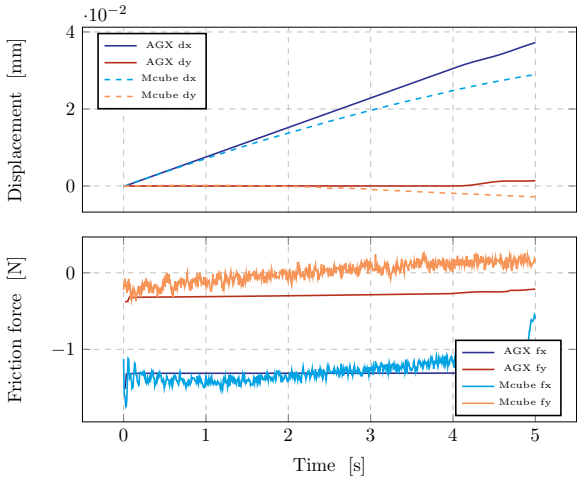
(a) Displacement and friction forces.



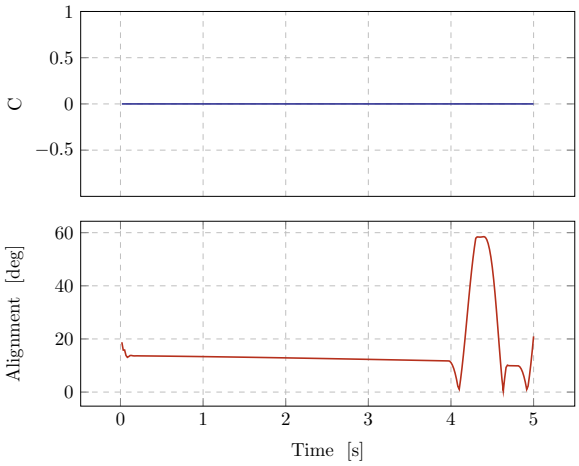
(b) Coulomb cone and alignment error.

Figure 4.5: Split solver on a 40° push with contact at the middle of the side.





(a) Displacement and friction forces.



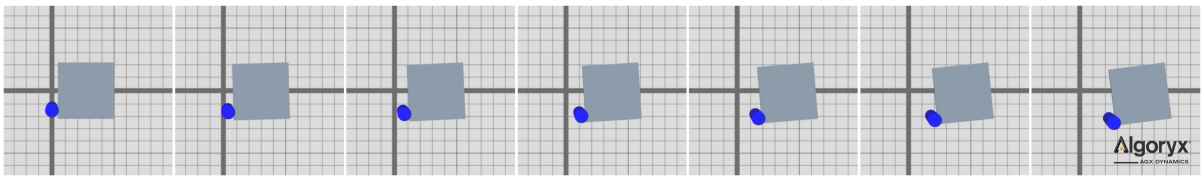
(b) Coulomb cone and alignment error.

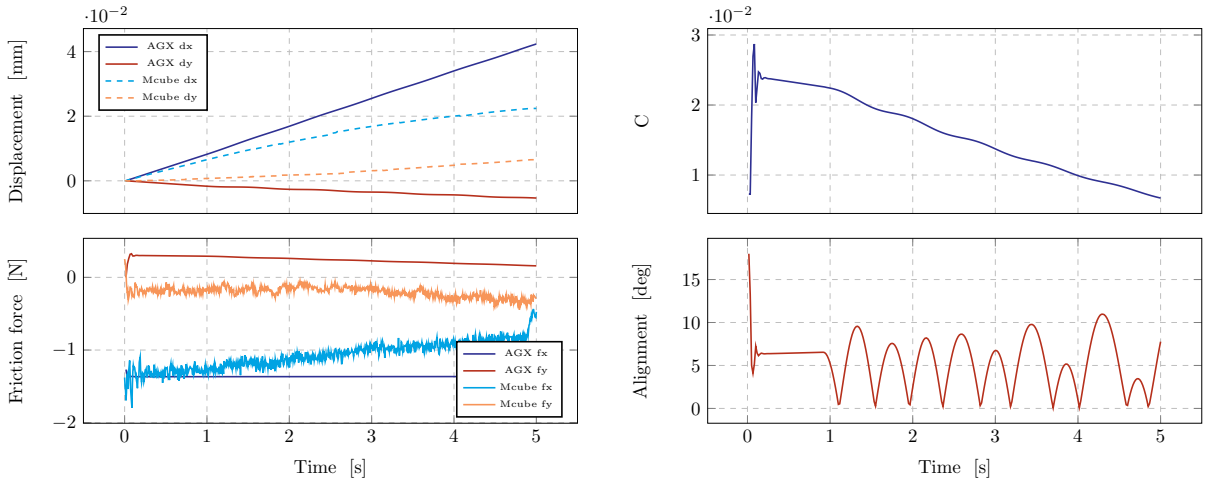
Figure 4.6: Direct & Iterative solver on a 40° push with contact at the middle of the side.

On the other angled push, the contact point is changed from 0.5 to 0.8, which means

that the object is being pushed at the corner. Together with the push direction, this should further enforce a rotation of the object. The results from the Direct and the Iterative solver are illustrated in Figure 4.7 and 4.8. Results from the other solvers on these pushes are found in Appendix A.2.

All solvers manage to rotate the object to some extent, especially when the Iterative projected cone friction model can be used. That is when using solver types that do not include direct solves.

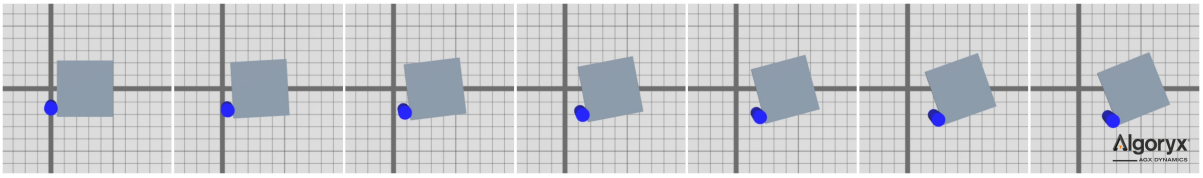


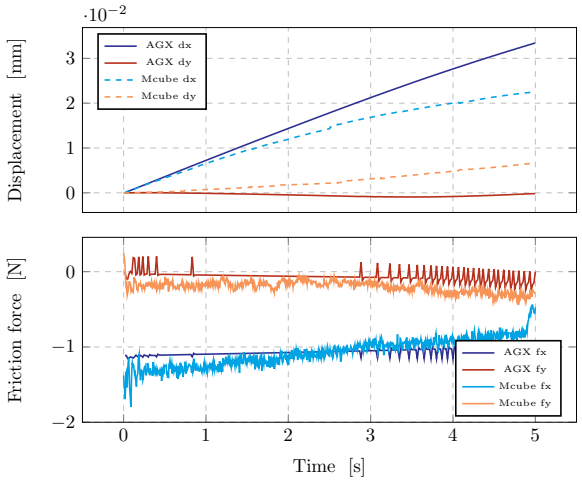


(a) Displacement and friction forces.

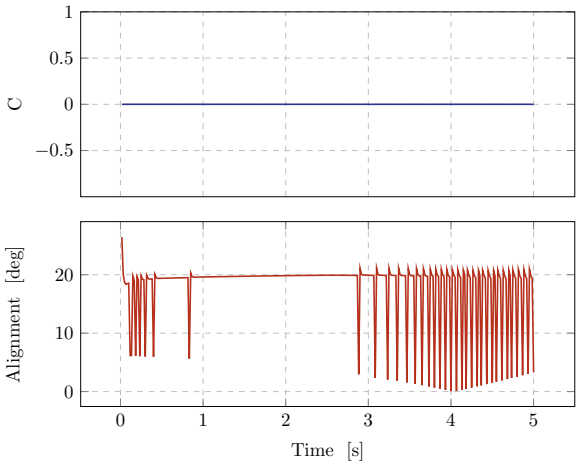
(b) Coulomb cone and alignment error.

Figure 4.7: Direct solver on a $-20°$ push at contact point 0.8.





(a) Displacement and friction forces.



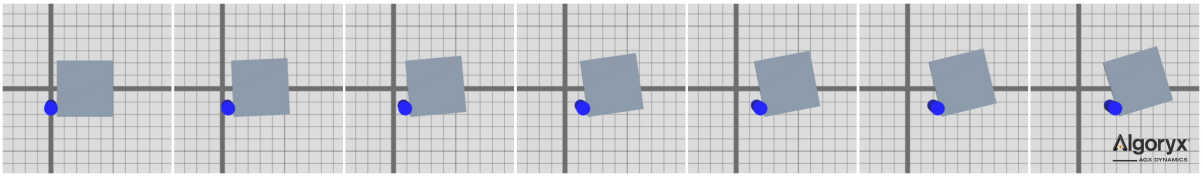
(b) Coulomb cone and alignment error.

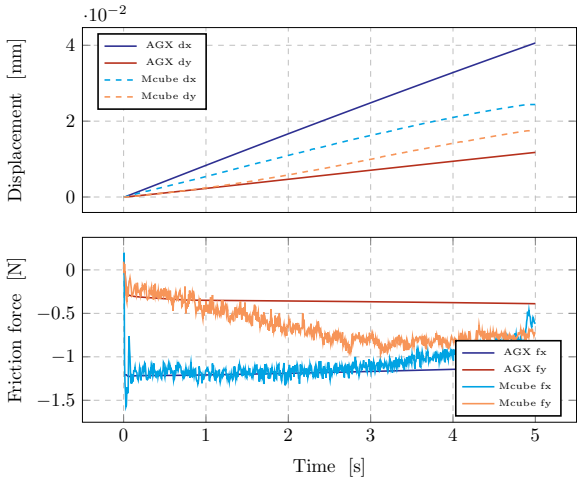
Figure 4.8: Iterative solver on a $-20°$ push at contact point 0.8.

4.3.3 The hardest case

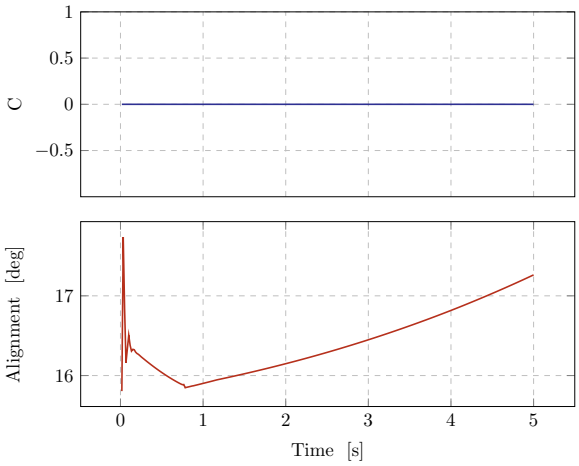
Out of the four pushes tested, this one proved to be the most complicated for the solvers to accurately match the experimental data. The push is directed completely straight in the x-direction but with the contact point at 0.8, close to the corner of the object.

As seen in Figure 4.10 the Direct solver pushes the object completely straight since there is no change in the y-direction. The friction forces are opposing any possible rotation of the object.



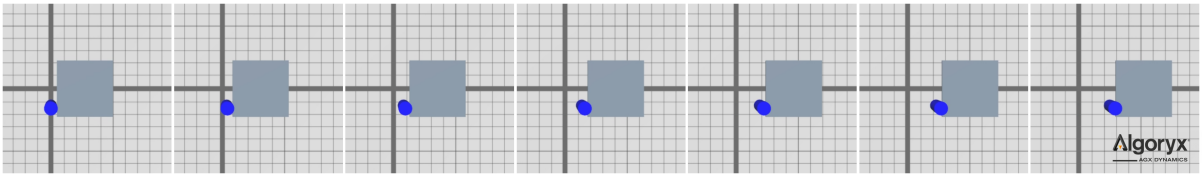


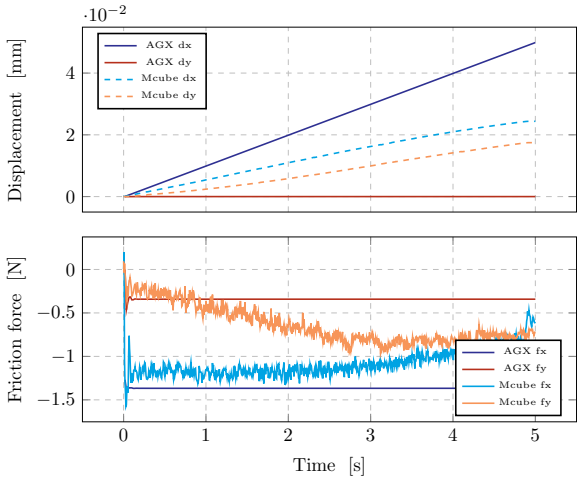
(a) Displacement and friction forces.



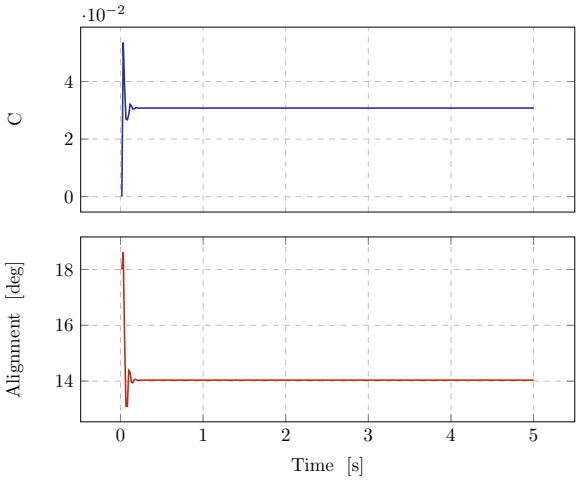
(b) Coulomb cone and alignment error.

Figure 4.9: Split solver on a straight push at contact point 0.8.





(a) Displacement and friction forces.



(b) Coulomb cone and alignment error.

Figure 4.10: Direct solver on a straight push at contact point 0.8.

4.3.4 Summary and discussion

The first push tested in this simulation experiment was completely straight, sliding on top of an isotropic surface and with an applied force aligned with the object's centre of mass. This was easy for all solver types to handle with zero or very small errors in alignment. Defining 'good' as being in compliance with the experimental results, this can be considered to be a good result.

However, as soon as the object is expected to rotate as a result of the push direction and contact point being changed, we do not always get that in the simulation. It is especially hard when direct solves are involved, which can be interpreted as a result of the box friction approximation. The friction budget is overestimated at the corner points of the box bounds which means that in the 45° direction the friction force can be up to 40% too large. That will definitely stop any attempts of rotation. This anisotropy in the contact plane comes with the box friction approximation and will always be aligned with the principal axes.

The Iterative solver causes no Coulomb errors but still gives some alignment error. It sometimes also struggles with fulfilling the constraint of keeping the object on top of the ground surface, which can be seen as spikes in the force measurement data. The Split solver is also doing well relative to the solvers containing direct solves and produces physical solutions regarding friction. It also shows the best match in displacement together with the Iterative solver. Alignment seems to always be more or less off from what it should, even when the Iterative projected cone friction model is used so it can not entirely be attributed to the box friction approximation.

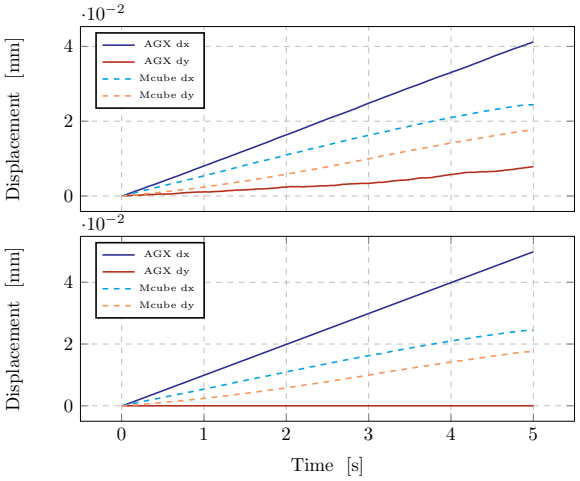
One trend that is showing in most of the tests done is that the object is always being pushed too far and not rotating enough. In the paper by Yu et al. [2] they have also presented results from simulations with a deterministic model. The trend there is also that the simulations push the object too far which means that friction forces are counteracting rotation. Their experiments show that friction expresses a degree of variability in many dimensions and suggests that frictional behaviour could be treated as a stochastic process. Maybe a level of structured uncertainty could be taken into account when further investigating how to solve for friction.

4.3.5 Planar pushing with improved box friction models

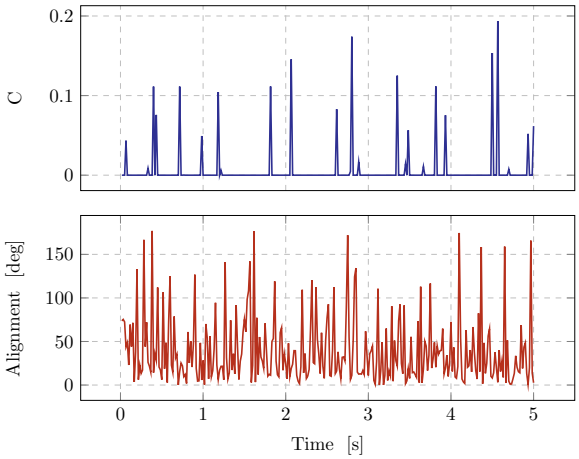
From previously presented results we can see that the box friction approximation has proven to cause issues in certain situations. Results on planar pushing with the developed implementations described in Section 3.5.1 and Section 3.5.2 are given here.

Randomized tangents

To avoid anisotropy in the tangent plane this new implementation rotates the alignment of the friction box around the contact normal by a random number $\theta \in [0, 360]$. The tests are performed on the so called 'hardest case' with contact point 0.8 and straight push direction.

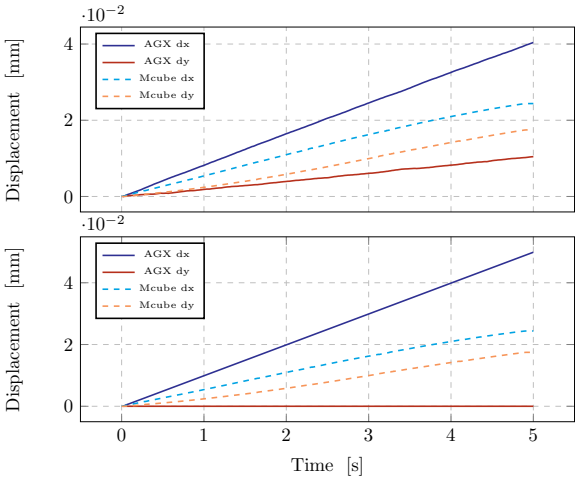


(a) Displacement.

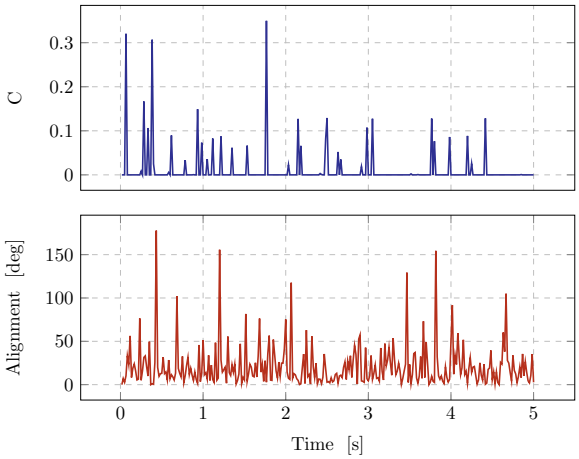


(b) Coulomb cone and alignment error

Figure 4.11: Direct solver on a straight push at contact point 0.8 with the Randomized tangents model. The lower panel on the left are results using the Scaled box friction model and the upper panel the one with randomized primary friction direction.



(a) Displacement.



(b) Coulomb cone and alignment error.

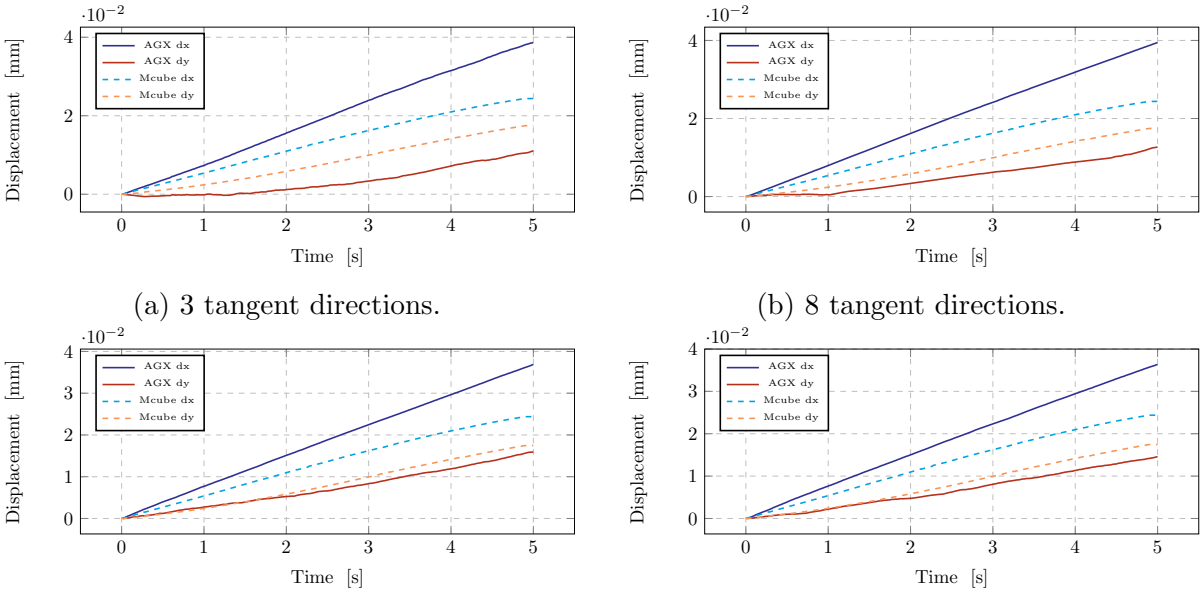
Figure 4.12: Direct & iterative solver on a straight push at contact point 0.8 with the Randomized tangents model.

Even though the box is shifted around in each time step the simulation can still handle the trivial case where the pusher is moving the object straight forward along the x-axis.

Since the direction of the forces in this implementation is going everywhere that data cannot be compared as of now.

Multi-directional tangent plane

By implementing custom constraints in AGX a workaround of having only two friction directions has been made possible. The Direct solver and box friction models made rotation impossible when pushing the object at a corner contact point, but with an increase in possible friction directions, some of the anisotropy in the tangent plane is removed. The straight push at contact point 0.8 is demonstrated in Figure 4.13 with a few different number of possible primary friction directions.



(a) 3 tangent directions.

(b) 8 tangent directions.

(c) 16 tangent directions.

(d) 64 tangent directions.

Figure 4.13: A straight push at contact point 0.8 with the multi-direction implementation of box friction and a variety of possible primary friction directions.

An important note is that if only two tangent directions are chosen, the object does not rotate as is expected from previous results when using the standard friction models.

4.4 Prehensile pushing

A prehensile pushing task is an action where a robot pushes an object against an external surface to achieve a different grasp. This can be utilized in manipulation as a strategy to increase the dexterity of robot hands. In the paper by Kolbert et al. [18] state-of-the-art contact models have been evaluated at predicting forces for a few in-hand manipulations involving prehensile pushing. One of them, linear pushing, has

been implemented and tested in AGX and compared with the experimental results presented in that paper. In their work, the environment is seen as an extra finger that acts on the object forcing it into a different grasp.

The experiment validated here is a prismatic-shaped object held in the grasp of a two-finger gripper and pushed against a flat surface along a straight line as seen in Figure 4.14. The object has dimensions $100 \times 25 \times 25$ mm, weight 200.5 g and is held by a gripper with circular fingertips with radius 20 mm. Contact forces are measured at the gripper's fingers, where the object is sliding, and against the wall surface of which it is being pushed. The object is being held with a gripping force of 20 N and pushed with a velocity of 10 mm/s. The procedure lasts for 1.5 s during which position and contact forces are recorded. These tests have been performed with the Direct and Split solver types. The Direct solver is given the default time step of 1/60 s. However, when iterative solves are involved the gripper struggles with keeping the object in the grasp. The Split solver with a time step of around 1/1000 s manages to perform the movement though, but due to slip the contact between the grasped object and the surface it is being pushed against is skewed and the contact points unstable. To force contact I have applied a small adhesion force of 0.5 N which is active throughout the simulation and resolves the issue.

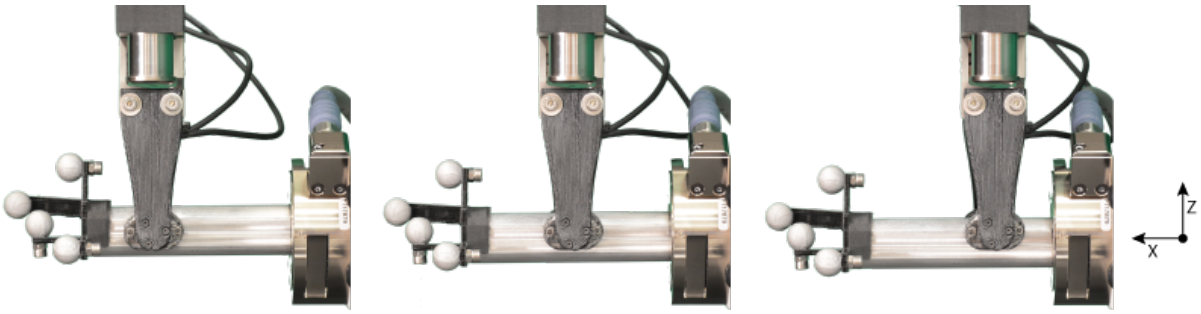
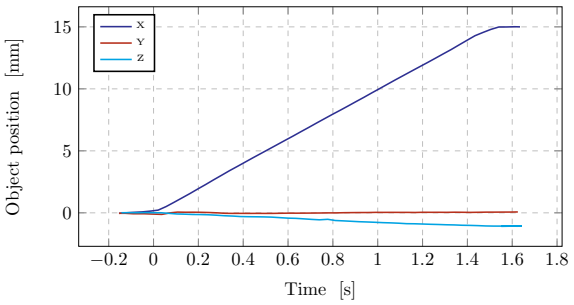


Figure 4.14: The experimental setup of the prehensile linear pushing. Image source: MCube Lab website [22].

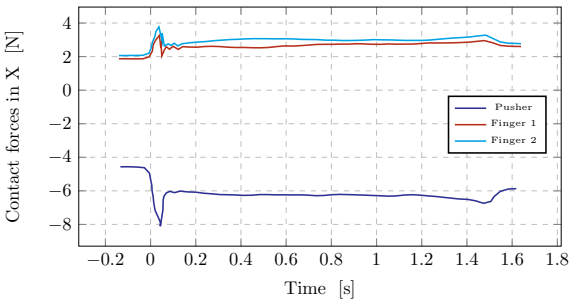
4.4.1 Uncertainties

The experimental results used in this benchmark lacks information regarding exactly which out of three possible objects the gripper is handling. They all have the same dimensions but different mass ranging from $94-200$ g. We also do not know what kind of tips the gripper's fingers are equipped with so the friction coefficient for the contact between object and gripper has been estimated from the results to be 0.15 using the Coulomb friction law.

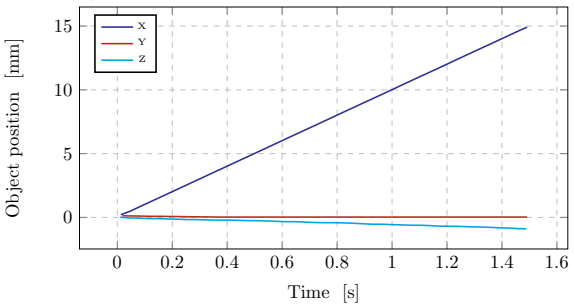
4.4.2 Results and discussion



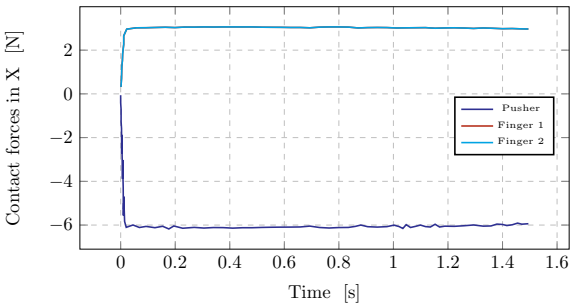
(a) Displacement from experiment.



(b) Contact forces from experiment.

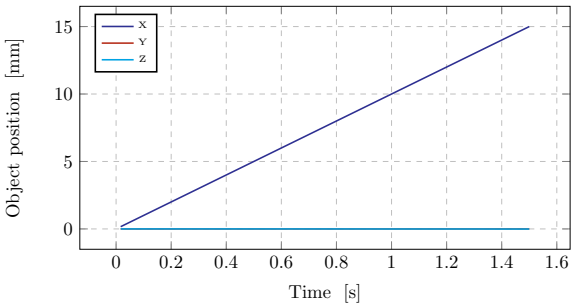


(c) Displacement from simulation.

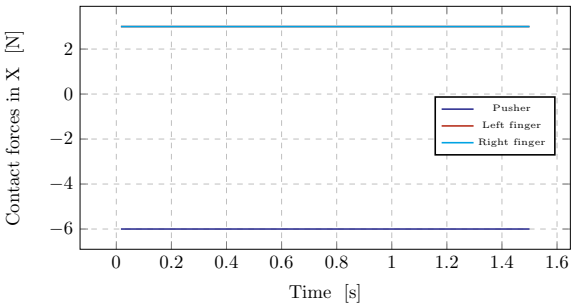


(d) Contact forces from simulation.

Figure 4.15: Results from MCube Lab on the prehensile linear pushing task with a gripping force of 20 N and pushing velocity of 10 mm/s. The data has been recreated from the plots presented in [18].



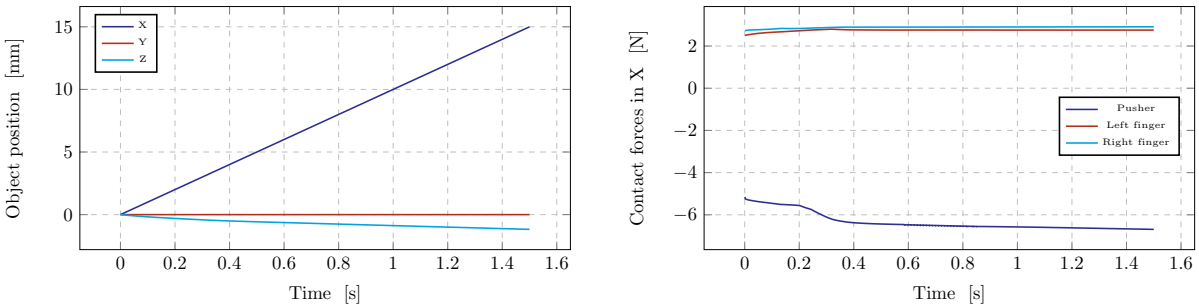
(a) Displacement.



(b) Contact forces.

Figure 4.16: AGX results with the Direct solver and a time step of 1/60 s.

One notable thing about the experimental results in Figure 4.15 is that the object is sliding downwards in the z-direction due to gravity as the gripper moves in the x-direction. MCube's own simulations manage to predict this movement but the Direct solver with the Scaled box friction model in AGX fails, as we can see in Figure 4.16. This also might be attributed to the Box friction model since the expected movement is towards the corner of the friction boxes at the contact points between the gripper fingers and the object. As we have learned from Section 4.3.4 the friction budget becomes too large at certain angles and the failure to slide down is probably another effect of that.



(a) Displacement.

(b) Contact forces.

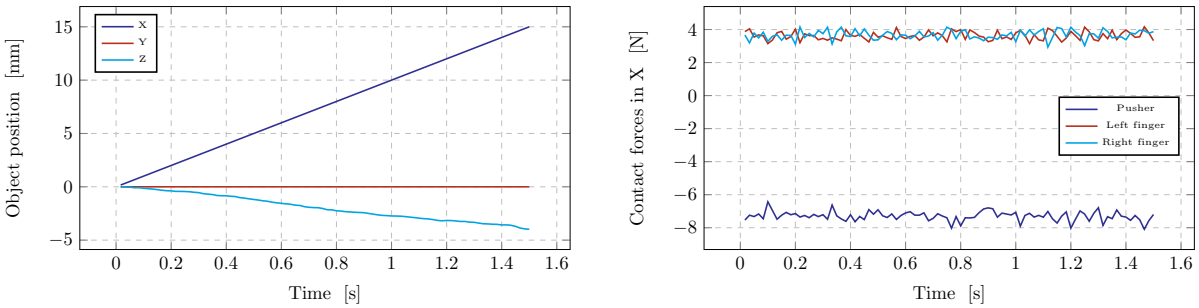
Figure 4.17: AGX results with the Split solver and a time step of 1/1000 s.

The Split solver on the other hand predicts this small sliding motion in the z-direction as can be seen in Figure 4.17. Here the Iterative projected cone friction model is used and there should be no anisotropy in the tangent planes. The normal force magnitude labelled 'Pusher' in Figure 4.17b includes the adhesion force set to keep a continuous contact between the object and the wall it is being pushed against, which is why it is slightly larger than the one in Figure 4.16b.

4.4.3 Prehensile pushing with improved box friction

Randomized tangents

Previously when using the Direct solver on the prehensile pushing task the object would not slide downwards as shown in experiments. Changing the friction model to the Randomized tangents the anisotropy seems to be gone but the sliding is instead exaggerated. The friction forces are also responsible for counteracting gravity and when the tangents are directed everywhere it fails to sufficiently hold the object as seen in Figure 4.18.

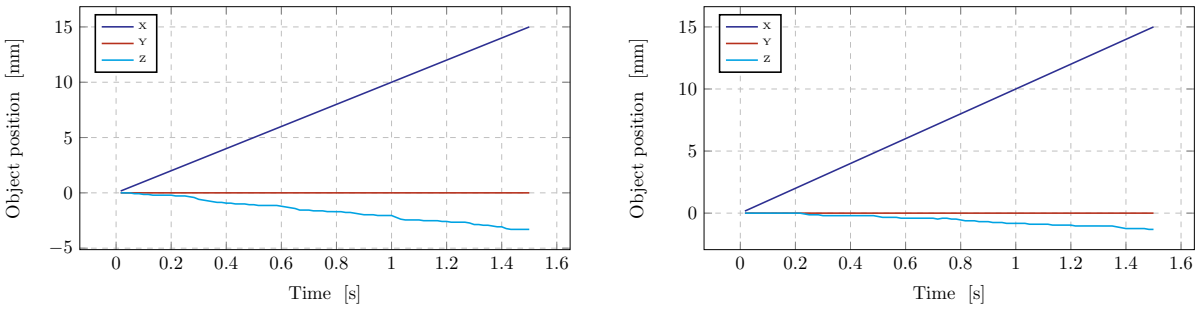


(a) Displacement. (b) Contact forces.

Figure 4.18: AGX results with the Randomized tangents friction model and Direct solver.

Multi-directional tangent plane

The multi-directional model has shown interesting possibilities for further development when using it on the planar pushing case. The tests on prehensile linear pushing are also promising but indicates that the object used by MCube Lab, whose results are presented in Figure 4.15, is actually one with a weight around 100 g and not the 200.5 g object previously used in simulation. This conclusion is drawn from what can be seen in Figure 4.19 where simulation results with both weights are presented.



(a) Displacement with mass = 200.5 g. (b) Displacement with mass = 100.5 g.

Figure 4.19: AGX results with the Multi-direction friction model and Direct solver with two different object weights.

4.5 Summary of improved box friction models

As seen in Figure 4.13a it is enough to add just one extra tangent direction to the Multi-direction friction model to initiate a rotation of the object during the push. More tangent direction improves the results even more compared to the MCube Lab data in terms of displacement. The sweet spot seems to be somewhere between 16-32 tangents since after that no further improvement has been noted. The computational time increases with an increase in tangent directions but for 32 tangents real-time

simulation is still possible. Unfortunately, contact forces could not be measured as of now with the Multi-direction friction model since it requires further work on the implementation. This is something that will be developed in the future.

The Randomized tangents model also shows an improvement in the displacement of the pushed object but has large errors relatively at a given time step. The alignment error is to be expected since the primary friction direction does not follow the slip velocity at all. However, the movement over time is certainly an improvement from the standard Box friction model. The current implementation of this model has some serious limitations though. The primary and secondary directions are set in AGX by creating a random rotation around a given axis that is defined by the user and then projected onto the desired tangent plane. This hard-coded feature is necessary to generalize and should depend on the current normal direction so that the model can be tested on a grasping task in the future.

Chapter 5

Conclusion

Looking back at the research problems stated in Section 1.1, the work of this thesis concludes that the friction models of AGX need further development to fully capture reality on situations where friction is a main part of the dynamics. The Direct solver in AGX is very powerful but limits the user to the box friction-based models which encounter challenges in some situations, especially when the motion of a system contains both rotation and translation.

The experimental benchmarks presented here were chosen because of their data quality and availability. More complex experimental manipulations involving gripping are desirable to further investigate AGX's abilities. Although the pushing task is simple, it still exposed some important areas where AGX could be improved. The new Randomized tangents friction model resolved some of the issues with anisotropy in the tangent plane when demonstrating it on the planar pushing task. Compared to using it on the prehensile linear pushing task it was not as suitable since the friction forces became directed in a way such that the gripper could not continuously uphold the object. What we can learn from it though is that more possible tangent directions in the contact plane can be beneficial and the Multi-direction friction models further demonstrate that. Out of the two improved models, I believe that the Multi-direction friction model has the most potential for further development.

In this thesis, I have investigated AGX Dynamics capabilities in terms of simulating physics that is important for grasping simulations. Challenges around friction have been detected and explained and ideas for improvement presented. Furthermore, AGX has been validated against experimental benchmark data which shows compliance but also exposes interesting weaknesses in certain situations.

5.1 Future Work

Friction has proved to be a hard nut to crack in simulation. The new models presented in this thesis are a start but need to be tested further on more complicated simulations to verify their worth. It is desirable to achieve isotropy in the tangent plane to avoid bias related to direction. More validation needs to be done to test for isotropy but also tests that check the direction of the tangent forces and their net magnitudes. In a grasping situation, a lot more contact points than in a pushing experiment will occur and the question is whether the new friction models are scalable without loss of performance. It is currently not known if the anisotropy effects worsen or improve simulations of complex grasping situations.

Bibliography

[1] Claude Lacoursière. "Ghosts and machines: regularized variational methods for interactive simulations of multibodies with dry frictional contacts". PhD thesis. Umeå University, 2007.

[2] Kuan-Ting Yu, Maria Bauza, Nima Fazeli, and Alberto Rodriguez. *More than a Million Ways to Be Pushed: A High-Fidelity Experimental Dataset of Planar Pushing*. 2016. arXiv: 1604.04038.

[3] OpenAI, Marcin Andrychowicz, Bowen Baker, Maciek Chociej, Rafal Józefowicz, Bob McGrew, Jakub W. Pachocki, Jakub Pachocki, Arthur Petron, Matthias Plappert, Glenn Powell, Alex Ray, Jonas Schneider, Szymon Sidor, Josh Tobin, Peter Welinder, Lilian Weng, and Wojciech Zaremba. "Learning Dexterous In-Hand Manipulation". In: (2018). arXiv: 1808.00177.

[4] David Orenstein. *People with paralysis control robotic arms using brain-computer interface*. URL: https://news.brown.edu/articles/2012/05/braingate2.

[5] Jennifer Andersson, Kenneth Bodin, Daniel Lindmark, Martin Servin, and Erik Wallin. "Reinforcement Learning Control of a Forestry Crane Manipulator". In: (2021). arXiv: 2103.02315.

[6] OpenAI, Ilge Akkaya, Marcin Andrychowicz, Maciek Chociej, Mateusz Litwin, Bob McGrew, Arthur Petron, Alex Paino, Matthias Plappert, Glenn Powell, Raphael Ribas, Jonas Schneider, Nikolas Tezak, Jerry Tworek, Peter Welinder, Lilian Weng, Qiming Yuan, Wojciech Zaremba, and Lei Zhang. "Solving Rubik's Cube with a Robot Hand". In: (2019). arXiv: 1910.07113.

[7] Angel Flores-Abad, Ou Ma, Khanh Pham, and Steve Ulrich. "A review of space robotics technologies for on-orbit servicing". In: *Progress in Aerospace Sciences* 68 (2014), pp. 1–26. ISSN: 0376-0421.

[8] Michael Danielczuk, Jingyi Xu, Jeffrey Mahler, Matthew Matl, Nuttapong Chentanez, and Ken Goldberg. "REACH: Reducing False Negatives in Robot Grasp Planning with a Robust Efficient Area Contact Hypothesis Model". In: *Int. S. Robotics Research (ISRR)* (2019).

[9] Matei Ciocarlie, Claire Lackner, and Peter Allen. "Soft Finger Model with Adaptive Contact Geometry for Grasping and Manipulation Tasks". In: 2007. DOI: 10.1109/WHC.2007.103.

[10] Ettore Pennestri, Valerio Rossi, Pietro Salvini, and Pier Paolo Valentini. "Review and comparison of dry friction force models". In: *Nonlinear Dynamics* 83 (Mar. 2016). DOI: 10.1007/s11071-015-2485-3.

[11] F. Barbagli, Antonio Frisoli, K. Salisbury, and M. Bergamasco. "Simulating human fingers: a Soft Finger Proxy Model and Algorithm". In: 2004. DOI: 10.1109/HAPTIC.2004.1287172.

[12] Jingyi Xu, Tamay Aykut, Daolin Ma, and Eckehard Steinbach. *6DLS: Modeling Nonplanar Frictional Surface Contacts for Grasping using 6D Limit Surfaces*. 2020. arXiv: 1909.06885.

[13] Aude Billard and Danica Kragic. "Trends and challenges in robot manipulation". In: *Science* 364.6446 (2019). DOI: 10.1126/science.aat8414.

[14] Ryan Elandt, Evan M. Drumwright, Michael Sherman, and Andy Ruina. "A pressure field model for fast, robust approximation of net contact force and moment between nominally rigid objects". In: (2019). arXiv: 1904.11433.

[15] Charles Bouchard, Matthieu Nesme, Maxime Tournier, Bin Wang, François Faure, and Paul Kry. "6D Frictional Contact for Rigid Bodies". In: *Graphics Interface*. Halifax, Canada, 2015. URL: https://hal.inria.fr/hal-01138834.

[16] C. Piazza, G. Grioli, M.G. Catalano, and A. Bicchi. "A Century of Robotic Hands". In: *Annual Review of Control, Robotics, and Autonomous Systems* 2.1 (2019). DOI: 10.1146/annurev-control-060117-105003.

[17] Nikhil Chavan-Dafle and Alberto Rodriguez. "Prehensile pushing: In-hand manipulation with push-primitives". In: *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2015. DOI: 10.1109/IROS.2015.7354264.

[18] Roman Kolbert, Nikhil Chavan-Dafle, and Alberto Rodriguez. *Experimental Validation of Contact Dynamics for In-Hand Manipulation*. 2017. arXiv: 1702.07252.

[19] Antonio Pérez-González, Carlos Fenollosa-Esteve, Joaquín Sancho-Bru, Tania Marin, Margarita Vergara, and Pablo-Jesús Rodríguez-Cervantes. "A modified elastic foundation contact model for application in 3D models of the prosthetic knee". In: *Medical engineering & physics* 30 (2008), pp. 387–98. DOI: 10.1016/j.medengphy.2007.04.001.

[20] Emma Sundling. "Validation Toolbox for a Physics Engine". MA thesis. Department of Physics, Umeå University, Jan. 1, 2016.

[21] D. E. Stewart and J. C. Trinkle. "An implicit time-stepping scheme for rigid body dynamics with inelastic collisions and Coulomb friction". In: *International Journal for Numerical Methods in Engineering* 39.15 (1996).

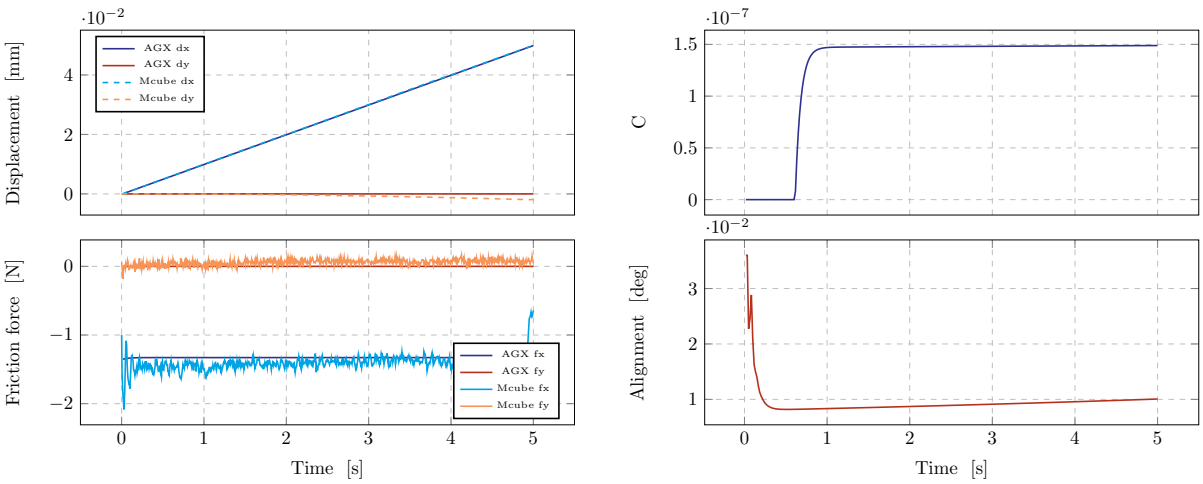
[22] Manipulation and Mechanisms Laboratory at MIT. *Push Dataset*. May 2021. URL: http://mcube.mit.edu/push-dataset/index.html.

[23] Claude Lacoursière, Martin Törnqvist, Hans A. Andersson, and Kenneth Bodin. *Regularized variational mechanics and sparse direct solvers for simulation of robot motion and grasping*. Algoryx Simulation, 2019.

Appendix A

Additional planar pushing results

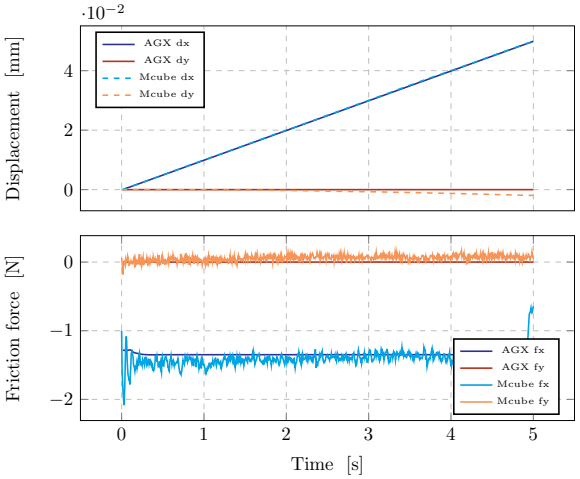
A.1 Straight pushing



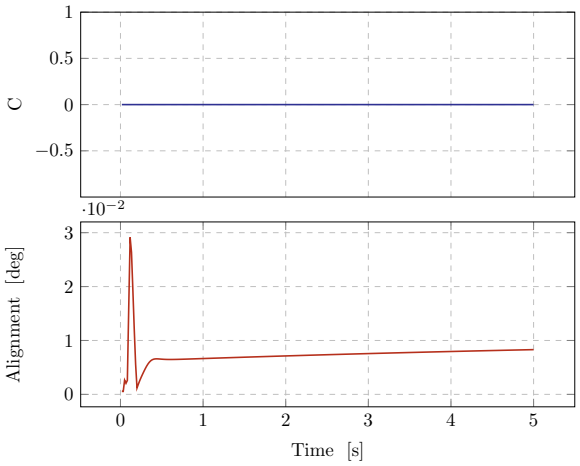
(a) Displacement and friction forces.

(b) Coulomb cone and alignment error.

Figure A.1: Split solver on a straight push with contact at the middle of the side.

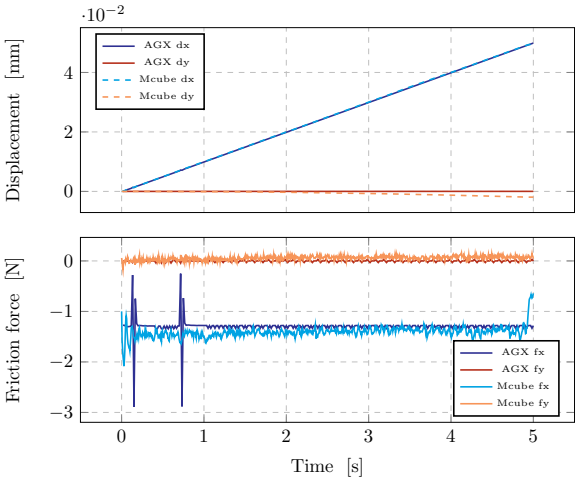


(a) Displacement and friction forces.

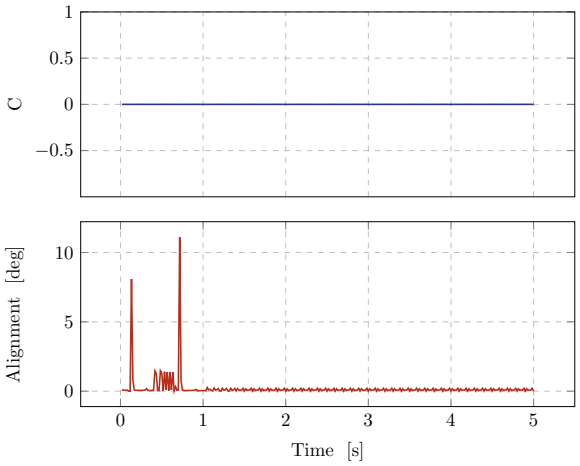


(b) Coulomb cone and alignment error.

Figure A.2: Direct & Iterative solver on a straight push with contact at the middle of the side.

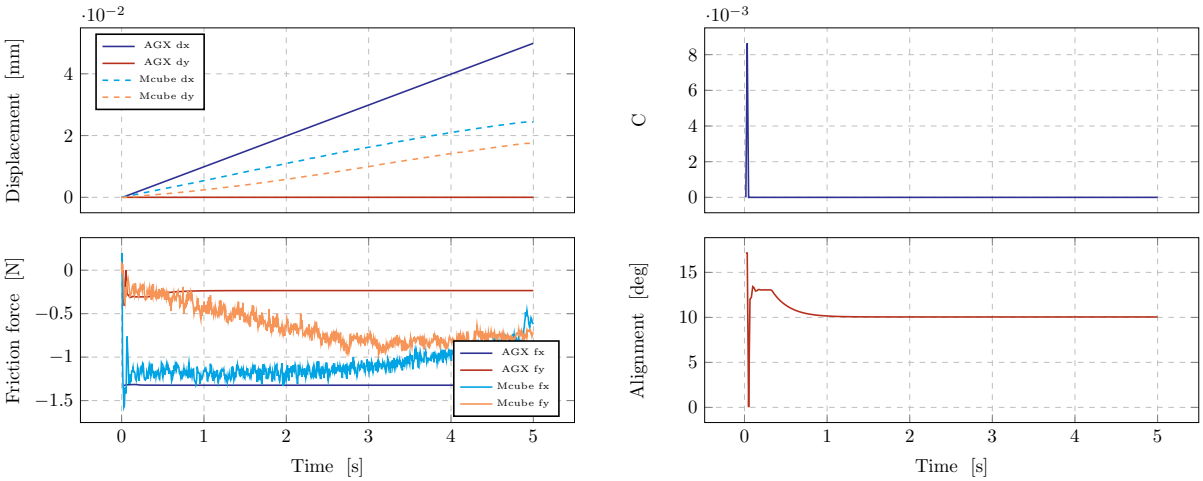


(a) Displacement and friction forces.



(b) Coulomb cone and alignment error.

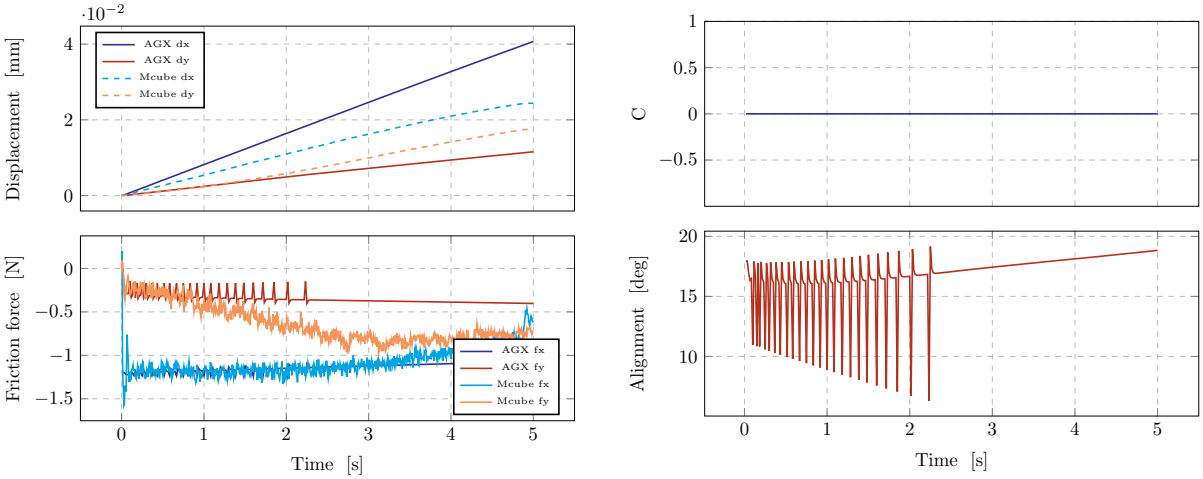
Figure A.3: Iterative solver on a straight push with contact at middle of the side. The solver is having issues with keeping the box entirely on top of the floor which is what causes the spikes.



(a) Displacement and friction forces.

(b) Coulomb cone and alignment error.

Figure A.4: Direct & Iterative solver on a straight push at contact point 0.8.

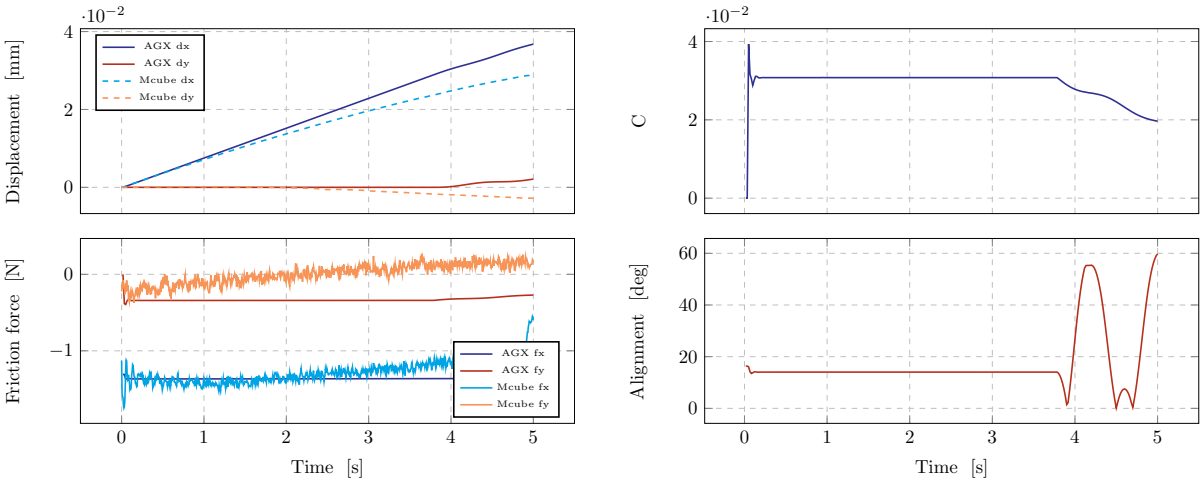


(a) Displacement and friction forces.

(b) Coulomb cone and alignment error.

Figure A.5: Iterative solver on a straight push at contact point 0.8.

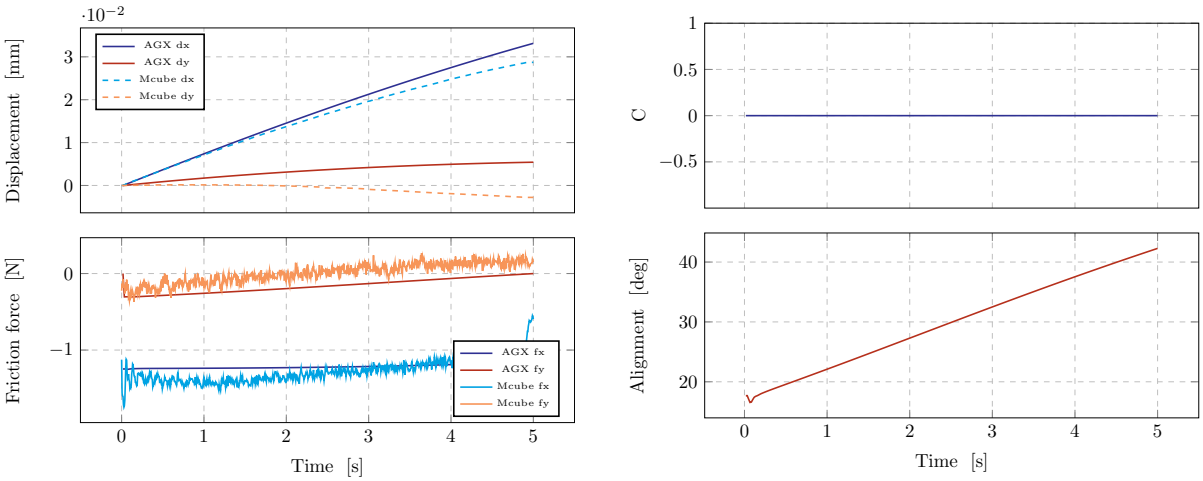
A.2 Angled pushes



(a) Displacement and friction forces.

(b) Coulomb cone and alignment error.

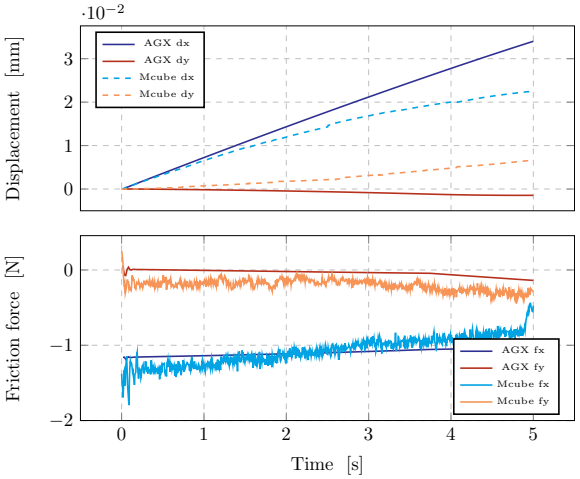
Figure A.6: Direct solver on a 40° push with contact at the middle of the side.



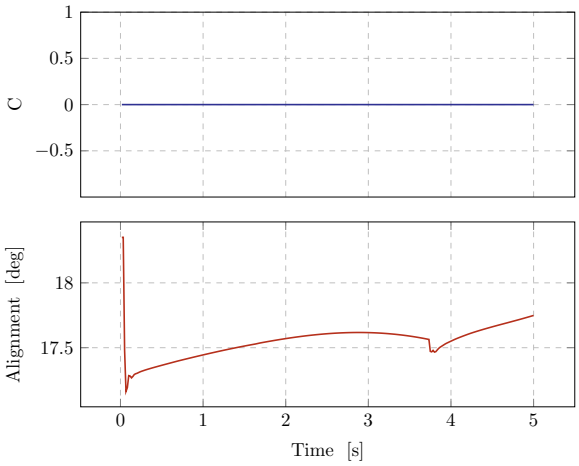
(a) Displacement and friction forces.

(b) Coulomb cone and alignment error.

Figure A.7: Iterative solver on a 40° push with contact at middle of the side.

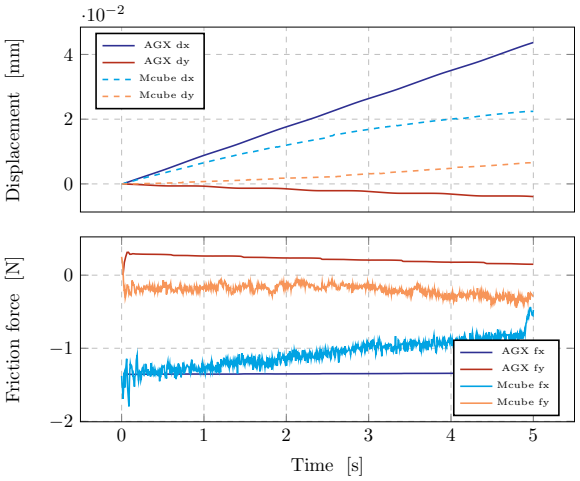


(a) Displacement and friction forces.

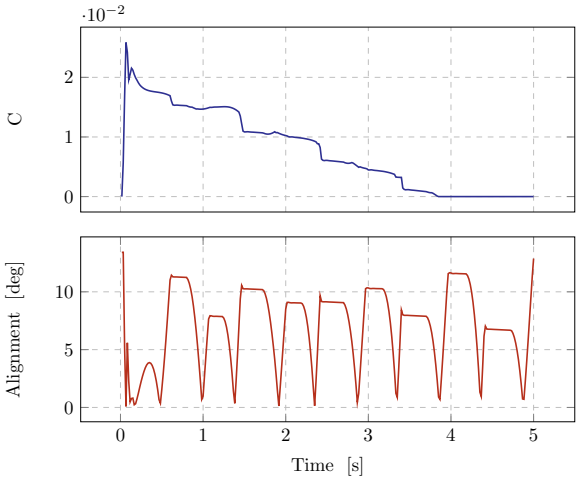


(b) Coulomb cone and alignment error.

Figure A.8: Split solver on a $-20°$ push at contact point 0.8.



(a) Displacement and friction forces.



(b) Coulomb cone and alignment error.

Figure A.9: Direct & Iterative solver on a $-20°$ push at contact point 0.8.