Examples of Dynamical Reduction Using Symmetry for Simple Mechanical Systems with Virtual Holonomic Constraints

Emrys Halbertsma

Supervised by
Dr. Christopher Nielsen

Presented to the Department of Physics & Astronomy on January 9th, 2023
in partial fulfillment of the requirements to PHYS 437A: Research Project.

Examples of Dynamical Reduction Using Symmetry for Simple Mechanical Systems with Virtual Holonomic Constraints

Emrys Halbertsma
Department of Physics & Astronomy,
University of Waterloo

Supervised by
Dr. Christopher Nielsen
Department of Electrical & Computer Engineering,
University of Waterloo

Abstract: A mechanical system can be made to appear holonomically constrained by actively enforcing a constraint through control. These so-called Virtual Holonomic Constraints (VHCs) have useful applications in robotics, such as maintaining set distances between agents or synchronizing the motions of joints. This project considers some examples of simple, 2$^{\text{nd}}$-degree underactuated mechanical systems with 1 degree of symmetry. Using Matlab, the mechanical systems are modelled and analyzed. The tangent spaces of their constraint manifolds are computed, along with the system's constrained dynamics, and then decomposed into an Ehresmann-like connection of Vertical and Horizontal Vector fields. The dynamical reduction technique can prove useful in simplifying the dynamics of robotic applications requiring the enforcement of a Virtual Holonomic Constraints.

Key terms: Virtual Holonomic Constraint, symmetry, geometric control, dynamical reduction.

Contents

List of Figures

Chapter 1

Introduction

1.1 Motivation

Modern robotic systems are designed as underactuated systems. These systems feature more degrees of freedom in the movement of the joints than there are actuators controlling their motions[3]. For robots with two or more degrees of underactuation, the dynamics may become highly complex. In fact, underactuated dynamics are often not even Euler-Langrange[2].

Figure 1.1: A Boston Dynamics four-legged robot demonstrates its ability to balance in 2021 [1]

It is therefore necessary to have a suite of fast, reliable methods for analyzing and controlling these underactuated robots. One approach is called dyamical reduction, wherein complex dynamics are simplified using mathematical techniques. Modeling a system to be with only a subset of its generalized coordinates, without compromising on the usefulness and accuracy of the model, allows for faster and more efficient computation and control.

1.2 Overview

This project explores the behaviour of some simple, underactuated, mechanical control systems endowed with symmetry subjected to virtual holonomic constraints (VHCs).

The chosen systems feature at least one Lagrangian symmetry, and two degrees of underactuation. Each actuator is made to enforce a VHC that respects the symmetry of the system–in other words, the system is controlled to maintain a particular configuration whose Lagrangian is invariant.

Each system is modelled in the Lagrangian formulation using Matlab. A suitable VHC is chosen, and then the constrained dynamics are derived using parametrization in terms of the VHCs. The constrained dynamics are also derived using Christoffel symbols–both methods should agree.

1.3 Notation

(1) $\mathrm{d}f_q$ is the Jacobian operator with respect to the coordinates q, applied to the vector function $f(q)$. Explicitly:

$$f := \vec{f}(\vec{q}), \tag{1.1}$$

$$\mathrm{d}f_q := \frac{\partial f^i}{\partial q^j} \hat{e}_i \cdot \hat{e}_j^\top . \tag{1.2}$$

(2) $A^\top$ is the transpose of the matrix A, explicitly $[A^\top]_{ij} = [A]_{ji}$.

(3) $\mathrm{Ker}\,(A)$ is the kernel, or nullspace, of a matrix A. Explicitly, the solution set $\{x \in \mathbb{R}^n : Ax = 0_n\}$.

(4) T_qQ is the tangent space of a point q on the manifold Q. By extension, TQ is the tangent bundle given by $TQ := \bigsqcup_{q \in Q} T_qQ$.

(5) Γ_{ij}^k is the $(i, j, k)^{\text{th}}$ Christoffel Symbol of the Riemannian metric, as defined in Equation 2.3.1.

(6) $\overset{\mathcal{C}}{\Gamma}{}_{ij}^k$ is the $(i, j, k)^{\text{th}}$ Christoffel Symbol of the constrained dynamics metric, as defined in Equation 2.3.2.

(7) $B^\perp$ is the left-annihilator of the matrix B, so that $B^\perp B = 0$.

Background

2.1 Modeling Mechanical Systems

2.1.1 Lagrangian Mechanics

Robotic systems may have complex movement ranges, combining various types of rotations, translations, or extensions. Attempting to model each joint of a robot as a coordinate of three-space (some subset of $\mathbb{R}^3$) can quickly become a mess of trigonometry. This motivates control systems engineers to instead model the robot's **configuration space** as a manifold Q, so that any configuration of the robot is an element of the Cartesian product of its joints' own configuration spaces.

Each coordinate q^i of a point $q \in Q$ is called a **generalized coordinate**. The number of coordinates corresponds to the robot's degrees of freedom.

A system can be modelled with the generalized coordinates $q \in Q$ using Lagrangian mechanics. This motivates the following definition.

Definition 2.1.1 (Smooth Simple Mechanical System [4])**.** A set $(Q, \mathbb{G}, P)$, where

i. Q is a smooth manifold called the configuration manifold. This is the manifold whose points consist of all possible coordinate configurations of the system.

ii. $\mathbb{G}$ is a smooth Riemannian metric on Q called the kinetic energy metric, typically defined as $\mathbb{G}(q, \dot{q}) = \frac{1}{2}\dot{q}^T D(q)\dot{q}$.

iii. P is a smooth potential function on Q. This may be a gravitational potential, but other potentials are common as well.

This definition will be expanded to a Smooth Simple Mechanical Control System in 2.2.2.

2.1.2 Holonomic Constraints

A system may be subject to constraints: a set of restrictions imposed on its possible velocity and positional configurations. **Holonomic contraints** are one class of constraint acting purely on the spatial (and time) coordinates of a system, but not on the velocity coordinates. Additionally, holonomic constraints can be written as a homogeneous equation.

A simple example is a point particle that is restricted to the surface of a sphere with radius R. Subject to this constraint, the point's spatial coordinates (x, y, z) are required to satisfy the equality

$$x^2 + y^2 + z^2 = R^2. \tag{2.1}$$

This is an example of a holonomic constraint, because we are able to define the constraint as a uniformly homogeneous function of the spatial coordinates (and time). Define the constraint h as:

$$h(x, y, z, t) := x^2 + y^2 + z^2 - R^2 \equiv 0. \tag{2.2}$$

Note that the constraint is independent of velocity, and is therefore *holonomic* by definition.

On the other hand, constraining that particle to the interior of the sphere would be an example of a **non-holomic constraint**.

$$x^2 + y^2 + z^2 < R^2 \implies h(x, y, z, t) := x^2 + y^2 + z^2 - R^2 < 0 \quad \text{and therefore} \quad h \neq 0. \tag{2.3}$$

While the constraint depends only on spatial coordinates, it cannot be expressed as an identically homogeneous function, which disqualifies it from being holonomic.

2.1.3 Virtual Holonomic Contraints

In this study, we are especially interested in artificially imposing holonomic constraints on mechanical systems by means of control. **Virtual holonomic constraints**, as they are called, are not “true” constraints on the system–the controller acts on the system’s actuators to make the system *appear* to be holonomically constrained. Of course, disabling the controller would allow the system to move in its natural, unconstrained fashion.

As a practical example, a Virtually Holonomic Constrained system could look like matching joint angles on a group of several identical robots such that they all walk in formation[5]. Suppose θ_i describes the i^{th} robot’s knee joint angle. If all robots walk in formation, all of their knee joints should be commanded to match some angle $\Theta(t)$. The VHC would therefore be written as,

$$h(\theta) := \sum_i \theta^i - \Theta(t) \equiv 0. \tag{2.4}$$

As a second example, modern vehicles are often equipped with advanced cruise control systems, which allow a driver to configure their vehicle to follow a set distance behind the vehicle ahead of them. This makes the two vehicles appear to be holonomically constrained (the norm of the difference between the vehicles’ position vectors is constant), and this constraint is artifically enforced by the cruise controller.

Maggiore and Consolini provide a formal definition for VHCs, as well as an overview of the conditions under which they can be imposed[5].

Definition 2.1.2 (Virtual Holonomic Constraint [5])**.** For a smooth simple mechanical control system $(Q, \mathbb{G}, P, F)$ as defined in Definition 2.2.2, a virtual holonomic constraint of order k is a smooth relation $h : Q \to \mathbb{R}^k$, $h(q) = 0$, and for all $q \in h^{-1}(0)$, and the constraint manifold

$$\mathcal{C} = \left\{(q, \dot{q}) : h(q) = 0, \mathrm{d}h_q \cdot \dot{q} = 0\right\} \tag{2.5}$$

is controlled invariant. That means, there exists a smooth feedback $\tau(q, \dot{q})$ that enforces the constraint, such that $\mathcal{C}$ is (asymptotically) invariant.

It is important to ensure a chosen VHC is well-defined and well-behaved.

Theorem 2.1.1 (Conditions for regular VHCs [5]). Let $h : Q \to \mathbb{R}^n$ be smooth, $\operatorname{rank} \mathrm{d}h_q = k$ for all $q \in h^{-1}(0)$. Then $h(q)$ is a **regular VHC of order** k if and only if $\forall q \in h^{-1}(0)$,

$$\dim \left[Im(D^{-1}(q)B(q)) \cap \operatorname{Ker}\left(\mathrm{d}h_q\right) \right] = n - 1 - k. \tag{2.6}$$

For the purpose of the dynamical reduction technique, we want select a set of VHCs that respect the symmetry of the given mechanical system.

2.1.4 Symmetry

This project considers mechanical systems endowed with one symmetry. A precise definition follows.

Definition 2.1.3 (Symmetry [4]). Let $(Q, \mathbb{G}, P)$ be a smooth simple mechanical system.

(a) A smooth left action Φ of a Lie group G on Q is a **symmetry** of the system if Φ is an isometry of $(Q, \mathbb{G})$ and F, P are Φ-invariant. In other words, the kinetic metric $\mathbb{G}$ and the potential and control forces are invariant under the action of Φ.

(b) A smooth vector field X on Q is an **infinitesimal symmetry** of the system is X is an infinitesimal isometry, and P is X-invariant.

A noteworthy extension of a above is that the Lagrangian $\mathcal{L}(\dot{q}, q)$ is also invariant under the action of Φ. This leads to Noether's Theorem, one of the key findings of 20th century physics.

Theorem 2.1.2 (Noether's Theorem (simplified) [4]). Let $(Q, \mathbb{G}, P)$ be a smooth simple mechanical system, and let $\gamma : I \to Q$ be a solution to its equations of motion, and J is the conserved quantity (momentum map) with respect to the symmetry. Then

(a) if $\Phi : G \times Q \to Q$ is a symmetry, then for all $\zeta \in g$,

$$\frac{\mathrm{d}}{\mathrm{d}t} \langle J_\Phi(t, \gamma'(t)), \zeta \rangle = 0. \tag{2.7}$$

(b) if X is an infinitesimal symmetry, then

$$\frac{\mathrm{d}}{\mathrm{d}t} J_X(\gamma'(t)) = 0. \tag{2.8}$$

2.2 Control Systems

2.2.1 Anatomy of a Control System

A **feedback control system** is a system that can be dynamically steered towards some desired target state, based on iterative feedback of its current state[6].

In control theory terminology, a **plant** is the part of the system that is being controlled–likely a robot in this context. The purpose of the control system is to push the robot towards a goal element or submanifold of the configuration manifold, known as a **reference signal** $r(t)$.

A **sensor** measures the variable of interest $y(t)$ on the plant. The sensor allows the system to perceive its current state, and then pass these measurements to the controller[6].

The **controller** itself is the brain of the system: its goal is to process the sensor inputs, and then output a command signal corresponding to the discrepancy between the reference state and the current state. This difference, $e(t) := r(t) - y(t)$, is called the **error**. The controller's outputted command signal $u(t)$ informs behaviour of the actuator(s) in the plant, resulting in the system's iterative convergence towards the desired state $r(t)$.

A controller is generally designed and tuned to meet a set of requirements[6], which can include minimizing $e(t)$ as quickly as possible, minimizing oscillations during convergence towards $r(t)$, maintaining stability under disturbances, maintaining stability on a large subset of Q, or minimizing the robot's energy cost of the control forces while still approaching $r(t)$.

Figure 2.1 shows the flow of information through a feedback control system.

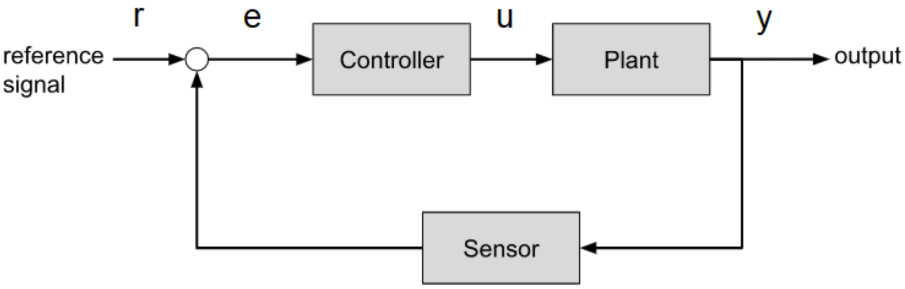

Figure 2.1: Block diagram of a simple feedback control system

2.2.2 Smooth Simple Mechanical Control Systems

Definition 2.1.1 must now be revised with the addition of another object: the control force. The configuration manifold Q is the set of possible states that the plant can occupy. Each actuator in the plant outputs a torque or force given by τ_i, mathematically modeled as a covector field on Q.

Definition 2.2.1 (Smooth Simple Mechanical Control System [4])**.** A set $(Q, \mathbb{G}, P, F)$, where

(a) Q is a smooth manifold called the configuration manifold. This is the manifold whose points consist of all possible coordinate configurations of the system.

(b) $\mathbb{G}$ is a smooth Riemannian metric on Q called the kinetic energy metric, typically defined as $\mathbb{G}(q, \dot{q}) = \frac{1}{2}\dot{q}^T D(q)\dot{q}$, where $D(q)$ is the intertia matrix.

(c) P is a smooth potential function on Q. This may be a gravitational potential, but other potentials are common as well.

(d) $F = \{\tau_1, \tau_2, \ldots, \tau_k\}$ is a set of covector fields on Q representing the control forces generated by each actuator on the system.

This is the only type of control systems considered through this project.

2.2.3 Modelling a Smooth Simple Mechanical Control System

Putting all the pieces together, a generalized differential equation for a simple mechanical control system is given by:

$$D\left(q\right)\ddot{q} + C\left(q,\dot{q}\right)\dot{q} + \nabla_q P\left(q\right) = B\tau \tag{2.9}$$

where each variable is defined as follows:

(a) q is the generalized coordinate vector at the point $q \in Q$ of the system, and $\dot{q}, \ddot{q}$ are the respective first and second time derivatives of q.

(b) $D(q)$ (also known as $M(q)$) is the inertia matrix describing the system's mass and intertia in each coordinate direction.

(c) $C\left(q,\dot{q}\right)$ is the Coriolis matrix.

(d) $P\left(q\right)$ is a scalar function whose gradient with respect to the generalized coordinate vector q describes the potential field of the system.

(e) B is a $(\dim q) \times (\dim \tau)$ matrix, which relates the degrees of freedom of the system to the number of control inputs. For example, if the system is characterized by 4 generalized coordinates and 2 actuators, $B \in M_{4\times 2}\left(\mathbb{R}\right)$.

(f) τ is the vector containing the elements of F (set of control inputs of the system described in Definition 2.2.2).

Constraints are expressed as a vector function, $h(q)$,

$$h(q) = \begin{pmatrix} h_1(q) \\ h_2(q) \\ \vdots \end{pmatrix} \equiv \vec{0}. \tag{2.10}$$

2.2.4 Control Error

Since the goal is to drive the holonomic constraints to zero through control, it makes sense to equate the VHC with the error:

$$e(q) := h(q). \tag{2.11}$$

From here, the corresponding time derivatives of the Error are found (omitting vector symbols for simplicity):

$$e = h(q); \tag{2.12}$$

$$\begin{aligned} \dot{e} &= \frac{\mathrm{d}}{\mathrm{d}t} h(q) \\ &= \mathrm{d}h\left(q\right)\dot{q}; \end{aligned} \tag{2.13}$$

$$\begin{aligned} \ddot{e} &= \frac{\mathrm{d}}{\mathrm{d}t}\left(\mathrm{d}h(q)\,\dot{q}\right) \\ &= \dot{q}^{\top}\left(\frac{\partial}{\partial q}\boldsymbol{\nabla}_q h(q)\right)\dot{q} + \mathrm{d}h(q)\,\ddot{q}. \end{aligned} \tag{2.14}$$

Where $\mathrm{d}h_q$ is the Jacobian operator on h, as defined in 1.

2.2.5 Control Law

By isolating for $\ddot{q}$ in Equation 2.14, we can set $\ddot{e}$ as a function of our control, τ. Then, we can choose the control, τ, so that it drives $\ddot{e} \to \vec{0}$. Starting from Equation 2.9,

$$\ddot{q} = D^{-1}(q) \Big(B\tau - C\left(q, \dot{q}\right) \dot{q} - \nabla_q P\left(q\right) \Big) . \tag{2.15}$$

Substitute Equation 2.15 into Equation 2.14), and assuming that $\ddot{e}$ is driven to zero, we can isolate for τ to obtain the **Control law**:

$$\begin{aligned} \tau &= \beta \left(-\frac{\partial h(q)}{\partial t} + \mathrm{d}h_q D^{-1} \left(C(q,\dot{q})\dot{q} + G(q) \right) - K_P e - K_D \dot{e} \right); \\ \text{where} \quad \beta &:= \Big(\mathrm{d}h_q D^{-1} B \Big)^{-1} . \end{aligned} \tag{2.16}$$

Here, a PD-style controller is used. These are easy to tune and implement, and have stood the test of time across industries[6]. The parameters K_P, K_D are the proportional and derivative controller parameters. These are chosen by the control system designer to meet whichever requirements are defined for satisfactory performance.

2.3 Concepts of Differential Geometry

While the natural basis elements of $\mathbb{R}^n$ are invariant throughout $\mathbb{R}^n$, this is not generally the case on manifolds.

Definition 2.3.1 (Christoffel Symbol)**.** Christoffel symbols are real numbers that describe how the basis element of a manifold transforms at each point on the manifold.
The $(i, j, k)^{\text{th}}$ Christoffel Symbol of the second kind is a real number $\Gamma^k_{ij} : Q \to \mathbb{R}$ defined by:

$$\Gamma^k_{ij} := \frac{1}{2} \sum_{l=1}^{n} D^{-1}_{kl} \left(\frac{\partial D_{jl}}{\partial q_i} + \frac{\partial D_{il}}{\partial q_j} - \frac{\partial D_{ij}}{\partial q_l} \right), \tag{2.17}$$

$$\text{for} \quad i, j, k \in \{1, 2, \ldots, n\} . \tag{2.18}$$

Here, D_{ij}, D^{-1}_{ij} denote the $(i, j)^{\text{th}}$ matrix elements of the inertia matrix $D(q)$ and its inverse, respectively.

The constrained dynamics have their own respective Christoffel symbols, which account for the parametrization $q = \varphi(s)$.

Definition 2.3.2 (Christoffel Symbols of the Constrained Connection)**.** The $(i, j, k)^{\text{th}}$ Christoffel Symbol corresponding to the constrained connection $\overset{\mathcal{C}}{\nabla}$ is a real number $\overset{\mathcal{C}}{\Gamma}{}^k_{ij} : Q \to \mathbb{R}$ defined by:

$$\overset{\mathcal{C}}{\Gamma}{}^k_{ij} := \sum_{a=1}^{n} \Big[(B^\perp D \mathrm{d}\varphi_s)^{-1} B^\perp D \Big]_{ka} \left(\frac{\partial^2 \varphi^a}{\partial s^i \partial s^j} + \left(\frac{\partial \varphi}{\partial s^i} \right)^\top \Gamma^a \left(\frac{\partial \varphi}{\partial s^j} \right) \right) q = \varphi(s) \Big| , \tag{2.19}$$

$$\text{for} \quad i, j, k \in \{1, 2, \ldots, n - m\} . \tag{2.20}$$

Here, D_{ij}, D_{ij}^{-1} denote the $(i, j)^{\text{th}}$ matrix elements of the inertia matrix $D(q)$ and its inverse, respectively.

Chapter 3

Methodology

3.1 Method of Dynamical Reduction

McCarthy et al. presents a Noether-like theorem for dynamical reduction[2]. The simplified steps for the reduction are outlined as follows, based on the work of McCarthy and Nielsen[7].

3.1.1 Parametrization of the Constraint Set

First, define a parametric function Ψ, that parametrizes the n-dimensional surface $\mathcal{C}$ defined by the chosen VHCs.

$$\Psi : X \to \mathcal{C} \tag{3.1}$$

$$\Psi : (s^1, s^2, \dots, s^n) \in X \mapsto q \in \mathcal{C}, \tag{3.2}$$

so that the tangent space of the constraint manifold is equal to the image of the Jacobian of the parametrization:

$$T_{q=\Psi(s)}\mathcal{C} = Im(\mathrm{d}\Psi_s). \tag{3.3}$$

A system with n degrees of underactuation requires the same number of parameters.

3.1.2 Computing the Constrained Dynamics

Next, revisit the general expression for the unconstrained system dynamics (Equation 2.9) to compute the constrained dynamics. The dynamics for each given point $q \in \mathcal{C}$ can be parametrized with the corresponding initial conditions,

$$x_0 = \begin{pmatrix} q \\ \dot{q} \\ \ddot{q} \end{pmatrix} = \begin{pmatrix} \Psi(s) \\ \mathrm{d}\Psi_s \dot{s} \\ \dot{s}\frac{\partial}{\partial s}\Psi_s \dot{s} + \mathrm{d}\Psi_s \ddot{s} \end{pmatrix}. \tag{3.4}$$

Equation 3.5 is converted purely in terms of s-coordinates, which are generally easier to work with than coordinates of a manifold. Use the left annihilator $B^{\perp}$ to obtain a homogeneous expression on the right hand side:

$$B^{\perp}\left(D(q)\ddot{q} + C(q,\dot{q})\dot{q} + \nabla_q P\right)\Big|_{x_0} = B^{\perp}B\tau\Big|_{x_0} \tag{3.5}$$

$$\left(B^{\perp}D(q)\ddot{q} + B^{\perp}C(q,\dot{q})\dot{q} + B^{\perp}\nabla_q P\right)\Big|_{x_0} = 0. \tag{3.6}$$

After performing the algebraic steps to solve for $\ddot{s}$, we have n second-order ODEs describing the constrained dynamics, giving $2n$ degrees of freedom.

$$\begin{aligned}
\ddot{s}^1 &= f_1(\dot{s}^1, \dots, \dot{s}^n, s^1, \dots, s^n), \\
\ddot{s}^2 &= f_2(\dot{s}^1, \dots, \dot{s}^n, s^1, \dots, s^n), \\
&\vdots \\
\ddot{s}^n &= f_n(\dot{s}^1, \dots, \dot{s}^n, s^1, \dots, s^n).
\end{aligned} \tag{3.7}$$

Note, however, that this project particularly focuses on second-degree ($n = 2$) underactuated systems, so the constrained dynamics tend to have only 4 degrees of freedom.

Alternative approach: There is also a coordinate-free approach to computing the constrained dynamics using Christoffel Symbols. The Christoffel Symbols of the Constrained dynamics are given by Equation 2.3.2. The computed values are in fact equivalent to Equation 3.7 above[7]. For a system with n degrees of underactuation:

$$\ddot{s}^k = -\sum_{i=1}^{n}\sum_{j=1}^{n} \overset{\mathcal{C}}{\Gamma}{}^k_{ij}(s)\dot{s}^i\dot{s}^j - \vec{e}_k^{\top}\left(B^{\perp}D\mathrm{d}\Psi_s\right)^{-1} B^{\perp}\nabla_q P\Big|_{q=\Psi(s)} \tag{3.8}$$

For a system with 2 degrees of underactuation, the result simplifies to:

$$\begin{aligned}
\ddot{s}^1 &= -\overset{\mathcal{C}}{\Gamma}{}^1_{11}(s)\dot{s}^1\dot{s}^1 - \overset{\mathcal{C}}{\Gamma}{}^1_{12}(s)\dot{s}^1\dot{s}^2 - \overset{\mathcal{C}}{\Gamma}{}^1_{21}(s)\dot{s}^2\dot{s}^1 - \overset{\mathcal{C}}{\Gamma}{}^1_{22}(s)\dot{s}^2\dot{s}^2 - \vec{e}_1^{\top}\left(B^{\perp}D\mathrm{d}\Psi_s\right)^{-1} B^{\perp}\nabla_q P\Big|_{q=\Psi(s)}, \\
\ddot{s}^2 &= -\overset{\mathcal{C}}{\Gamma}{}^2_{11}(s)\dot{s}^1\dot{s}^1 - \overset{\mathcal{C}}{\Gamma}{}^2_{12}(s)\dot{s}^1\dot{s}^2 - \overset{\mathcal{C}}{\Gamma}{}^2_{21}(s)\dot{s}^2\dot{s}^1 - \overset{\mathcal{C}}{\Gamma}{}^2_{22}(s)\dot{s}^2\dot{s}^2 - \vec{e}_2^{\top}\left(B^{\perp}D\mathrm{d}\Psi_s\right)^{-1} B^{\perp}\nabla_q P\Big|_{q=\Psi(s)}.
\end{aligned} \tag{3.9}$$

3.1.3 Sufficient Conditions for the Dynamical Reduction

McCarthy provides sufficient conditions to apply the dynamical reduction technique are given in the article under the Assumption below.

Assumption ([2])**.** For a manifold $\mathcal{C}$ of dimension 2, there exists a “horizontal” sub-bundle $H\mathcal{C}$ of $T\mathcal{C}$, with $\operatorname{rank} H\mathcal{C} = 1$, satisfying:

(a) $H\mathcal{C}$ is complementary to the vertical sub-bundle $V\mathcal{C}$, so that $T\mathcal{C} = H\mathcal{C} \oplus V\mathcal{C}$

(b) $\overset{\mathcal{C}}{\nabla}_V V \in \Gamma^{\infty}(H\mathcal{C})$;

(c) For all $X, Y \in \Gamma^{\infty}(H\mathcal{C})$, $\overset{\mathcal{C}}{\nabla}_X Y \in \Gamma^{\infty}(H\mathcal{C})$, i.e., $H\mathcal{C}$ is autoparallel.

In essence, we assume it is possible to separate the tangent space of the configuration manifold $\mathcal{C}$ into the direct sum of a so-called “Vertical Vector Field” and a “Horizontal Vector Field”. This draws similarities to an Ehresmann Connection (Definition **??**).

3.1.4 Assigning a Vertical Vector Field

The Vertical component is defined by the direction of the symmetry of the system.

Let Q be the configuration space of a simple mechanical control system. Let the left action of the Lie group G, $\Phi : Q \times G \to Q$, be a symmetry of the system.

$$\Phi : q \in Q, g \in G \mapsto q + g \in Q. \tag{3.10}$$

Then the direction of the symmetry with respect to G is,

$$V(q) := \frac{\partial \Phi}{\partial g}, \tag{3.11}$$

and we set this to be the Vertical Vector Field. Note that we expect $V(q) \in T_q\mathcal{C}$. We expect that together, $\mathrm{Span}\big(V(q), H(q)\big)$ will form a basis of the tangent space of the constraint manifold.

3.1.5 Assigning a Horizontal Vector Field

The Horizontal component is chosen based on the given Virtual Holonomic Constraints. One option is to take $H(q) = \nabla_q h(q)$. Here, it is important to validate that the Assumptions given in 3.1.3 apply, and to ensure the horizontal and vertical fields together span the tangent space $T_q\mathcal{C}$.

We can now define a vertical projection operator σ_q^V, which outputs purely the verical component of the field for a given $X \in T_q\mathcal{C}$. We define X as the direct sum of the two fields $X = \alpha V + \beta H$, and so the projection simply outputs the α coefficient. This reduces the system dynamics to only α (while H is constant), if X is parallel transported along a given curve on the surface of $T_q\mathcal{C}$.

3.2 Controller Design

After selecting a mechanical system, the next step is to design a controller which can stabilize the system to the constraint manifold defined by the VHCs. The standard approach involves setting up a differential equation to model the dynamics of the system, then choosing a feedback control to minimize the error, and tuning the parameters until an optimal (or at least satisfactory) performance is attained[6]. The control law is already given in Equation 2.16 for a PD-style controller.

3.2.1 Well-Definedness of Controller Enforcing VHC

Before implementing a controller, it is important to verify that a VHC is *regular* (Theorem 2.1.3), as this is a necessary condition for the constrained system to be stabilizable by a controller.

The program checks this using the method given by Maggiore in the proof of the theorem ([5, Prop. 3.2]).

```matlab
function [N, D] = is_well_defined(h, q, M, B)
    dhq = jacobian(h, q);
    span_of_kernel = null(dhq); % 3x2
    V = [span_of_kernel, M \ B]; % 3x3
    [N, D] = numden(det(V));
end
```

If the outputted numerator `N` is zero, the controller is well-defined.

3.2.2 Controller Gains

The controller can be tuned optimally using the `LQR` function in Matlab. However, for the purposes of this project, a satisfactory performance was achieved using simple gains of $K_P = K_D = 5.0$. This led to convergence of the virtual holonomic constraints within $\Delta t = 10$ for most cases when initial conditions were at least somewhat close to the constraint set $\mathcal{C}$.

3.3 Simulation

3.3.1 Overview

Matlab was chosen for its extensive documentation, and for its popularity in industrial control systems design. This project made use of Matlabs's Symbolic Toolbox and Control System Toolbox.

Simulations of the systems described in Section **??** were conducted in Matlab. As mentioned, some previous work had been done by a former student, so this was used as a starting point. In particular, the computation of Christoffel symbols and parametrized constraint dynamics methods are built on this previous work.

The program accepts a configuration file for a simple mechanical system definition, its coordinate relationships, and a set of VHCs. The system dynamics are then computed and modeled, and then simulated over a time interval.

3.3.2 Program Architecture

1. The user starts at `new_system_template.m`. When prompted, the user enters the system name, and a new directory and configuration file are generated.

2. Next, the user navigates to the new configuration file to enter the system parameters.

3. The system can then be imported using `system_dynamics.m`, which converts the configured parameters into a model of the mechanical system. The relevant Lagrangian, inertia, coriolis, and actuator functions are saved in the system's subdirectory. This script makes use of Matlab's Symbolic Toolbox.

4. From there, `simulate_system.m` sets up the differential equation that we wish to solve. The script calls on a series of sub-functions to model the system term by term at each timestep of the simulation.

5. Finally, `plot_system.m` outputs visualizations of the system dynamics. These are saved in a subfolder with the same name as the system.

3.3.3 Simulating a Smooth Simple Control System

The program uses Matlab's `ode45` solver, an implementation of the Range-Kutta algorithm with varying time steps[8].

The solver iterates through a specified interval and timestep to solve Equation 2.9 corresponding to the instantaneous $D(q), C(\dot{q}, q), P(q)$,and B matrices. The system's behaviour is then corrected using the control signal τ to drive the system towards the constraint set defined by the VHCs.

Other required parameters include:

(a) PD controller constants K_P, K_D,

(b) Numerical values for segment lengths, masses, B-matrix, and gravitational constant

(c) A vector `q0` of initial positions and velocities of the coordinates.

The program outputs and saves a series of plots visualizing the system's behaviour.

3.3.4 Version Control

The project is stored on Dr. Nielsen's Gitlab instance. A new branch titled `fall2022-dev` contains the newly developed code.

Chapter 4

Results

4.1 Offset Robotic Arm

This section provides the complete worked example of the methodology, illustrating the process from selection of a mechanical system all the way to dynamical reduction.

This example is adapted from **??**, modified such that the actuated joint is offset from the centre of the block. This results in a coupling of the dynamics of the arm and the block–a much more interesting problem. Additionally, the actuator controlling the length of the arm is removed. This gives the system two degrees of underactuation, which is favourable for the dynamical reduction technique being explored.

Description Consider a massive body which rotates around a fixed point in space. A leg extends from the body, which can rotate relative to the body and change its length.

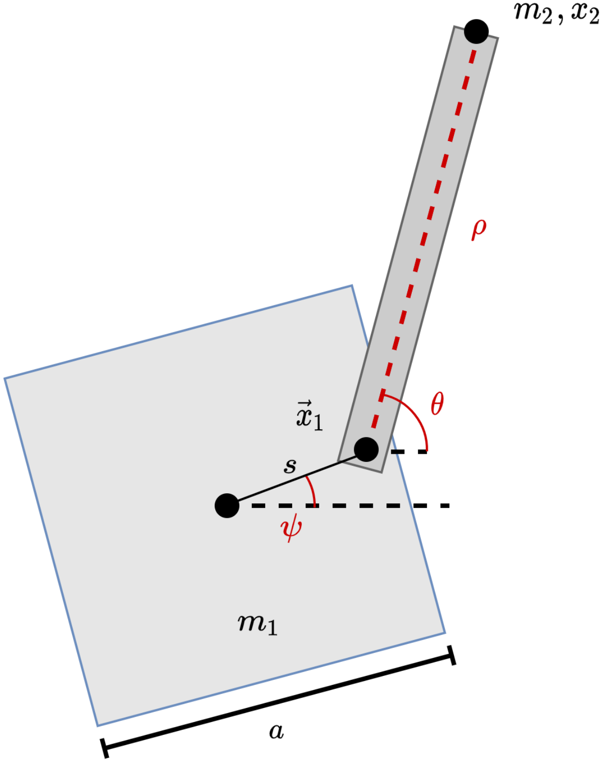

Figure 4.1: Offset robotic arm schematic diagram

(1) The smooth simple simple mechanical control system $(Q, \mathbb{G}, F, P)$ is given by:

- **Configuration space** $Q = \mathbb{R}_{>0} \times \mathbb{S}^1 \times \mathbb{S}^1$, with coordinates $q = (r, \theta, \psi)$
- **Kinetic metric** $\mathbb{G} = m(\mathrm{d}r \otimes \mathrm{d}r + r^2 \mathrm{d}\theta \otimes \mathrm{d}\theta) + J\mathrm{d}\psi \times \mathrm{d}\psi$
- **Actuation** $F = (\mathrm{d}\theta, \mathrm{d}\tau)$
- **Potential** In this example, there is no potential field: $\nabla P = (0, 0, 0)$. This is chosen to allow the existence of a symmetry in the system dynamics.

From the diagram of the system, it can be shown that the spatial coordinates are

$$\vec{x}_1 = s\cos\psi\hat{x} + s\sin\psi\hat{y}, \tag{4.1}$$

$$\vec{x}_2 = \vec{x_1} + \rho\cos\theta\hat{x} + \rho\sin\theta\hat{y}. \tag{4.2}$$

The velocities are given by,

$$\dot{\vec{x}}_1 = -s\dot{\psi}\sin\psi\hat{x} + s\dot{\psi}\cos\psi\hat{y}, \tag{4.3}$$

$$\dot{\vec{x}}_2 = \dot{\vec{x}}_1 + (\dot{\rho}\cos\theta - \rho\sin\theta\cdot\dot{\theta})\hat{x} + (\dot{\rho}\sin\theta + \rho\cos\theta\cdot\dot{\theta})\hat{y}. \tag{4.4}$$

(2) The system's Lagrangian depends on the kinetic energy of the solid body and of the point mass. There is no potential in this model.

$$\mathcal{L}(q, \dot{q}) = \frac{1}{2}\left(\frac{a^2}{12}m_1 + s^2 m_2\right)\dot{\psi}^2 + \frac{1}{2}m_2\left(\dot{\rho}^2 + \rho^2\dot{\theta}^2\right) + m_2 s\left(\dot{\rho}\dot{\psi}\sin(\theta - \psi) + \psi\rho\dot{\theta}\cos(\theta - \psi)\right). \tag{4.5}$$

(3) From Equation 4.5, the system's symmetry is found to be in the difference $\xi := \theta - \psi$. Define the left action of the Lie group $\mathbb{S}^1$ to be $\Phi : Q \times \mathbb{S}^1 \to Q$ to be the symmetry of the system, where the configuration space is $Q = \mathbb{R}^+ \times \mathbb{S}^1 \times \mathbb{S}^1$.

$$\Phi : \begin{pmatrix} \rho \\ \theta \\ \psi \end{pmatrix} \in Q, g \in G \mapsto \begin{pmatrix} \rho \\ \theta + g \\ \psi + g \end{pmatrix} \in Q. \tag{4.6}$$

We can see that for all $g \in G$, this symmetry has no effect on ξ:

$$\xi(q) := \theta - \psi, \tag{4.7}$$

$$\xi\left(\Phi(q, g)\right) = (\theta + g) - (\psi + g) = \theta + \psi + g - g = \xi(q). \checkmark \tag{4.8}$$

Since the Lagrangian's θ, ψ dependency can be expressed as a function of ξ, the Lagrangian is also invariant under the action of Φ. This means one of θ, ψ can be considered *cyclic*, which in turn means that there is an associated conserved quantity[9, 260].

(4) The next step is to choose well-defined Virtual Holonomic Constraints that respect the symmetry defined above.

The set of chosen VHCs $h(q)$ (Definition 2.1.3) constrain the system to a submanifold $\mathcal{C} \subset Q$, with

$$T_q\mathcal{C} := \mathrm{Ker}\left(\mathrm{d}h_q\right). \tag{4.9}$$

This tangent space can equivalently be defined using the Ehresmann connection. Take $V(q)$ as the Vertical Vector Field defined by the direction of the symmetry:

$$V(q) := \frac{\partial \Phi}{\partial g} = \frac{\partial}{\partial g}\begin{pmatrix}\rho \\ \theta + g \\ \psi + g\end{pmatrix} = \begin{pmatrix}0 \\ 1 \\ 1\end{pmatrix}. \tag{4.10}$$

Together with an appropriate Horizontal Vector Field $H(q)$ of our choosing, we can get,

$$T_q\mathcal{C} = \operatorname{Span}\left(\{\,\}\right)V(q), H(q). \tag{4.11}$$

Now, we must select VHCs (and thereby also a constraint manifold $\mathcal{C}$) such that Equation 4.9 holds true.

The robot arm system has 3 degrees of freedom and 1 actuator, equating to $3 - 1 = 2$ degrees of underactuation. Since we plan to use this actuator to enforce the VHC, the constraint manifold should have $\dim \mathcal{C} = 2$ remaining degrees of freedom.

Select the following VHCs:

$$h(q) := \theta - \psi \equiv 0, \tag{4.12}$$

$$\mathrm{d}h_q = \begin{pmatrix}\frac{\partial h}{\partial \rho} & \frac{\partial h}{\partial \theta} & \frac{\partial h}{\partial \psi}\end{pmatrix} = \begin{pmatrix}0 & 1 & -1\end{pmatrix}, \tag{4.13}$$

$$\operatorname{Ker}\left(\mathrm{d}h_q\right) = \left\{\begin{pmatrix}1 \\ 0 \\ 0\end{pmatrix}, \begin{pmatrix}0 \\ 1 \\ 1\end{pmatrix}\right\}. \tag{4.14}$$

Since $V(q)$ is seen in this basis (Equation 4.10), it is natural to choose

$$H(q) = \begin{pmatrix}1 \\ 0 \\ 0\end{pmatrix}, \tag{4.15}$$

and their direct sum defines the fiber bundle $T\mathcal{C} = TV \oplus TH$.

(5) The constrained dynamics are computed in two different methods. The first method involves a parametrization of the coordinates subject to the VHCs, and the second method uses Christoffel symbols.

The Christoffel Symbols (Definition 2.3.1) of the constraint manifold $\overset{\mathcal{C}}{\Gamma}{}^k_{ij}$ are computed in Matlab:

$$\begin{array}{ll} \overset{\mathcal{C}}{\Gamma}{}^1_{11} = 0 & \overset{\mathcal{C}}{\Gamma}{}^2_{11} = 0 \\ \overset{\mathcal{C}}{\Gamma}{}^1_{12} = \rho & \overset{\mathcal{C}}{\Gamma}{}^2_{12} = 0 \\ \overset{\mathcal{C}}{\Gamma}{}^1_{21} = \rho & \overset{\mathcal{C}}{\Gamma}{}^2_{21} = 0 \\ \overset{\mathcal{C}}{\Gamma}{}^1_{22} = -L - \rho & \overset{\mathcal{C}}{\Gamma}{}^2_{22} = 0 \end{array} \tag{4.16}$$

4.2 Spherical Pendulum

This example was used to motivate the Noether-like theorem in McCarthy's paper. The mechanical system as described in the article is modeled and simulated using the Matlab program.

Description A point mass m_1 on the xy-plane can be actuated in the radial and tangential components (ρ, θ). Attached to m_1 is a rigid massless rod of fixed length ℓ with another point mass m_2 at the end. The second mass m_2 is on or above the xy-plane, and is free to move in the azimuthal and tangential components (ϕ, ψ).

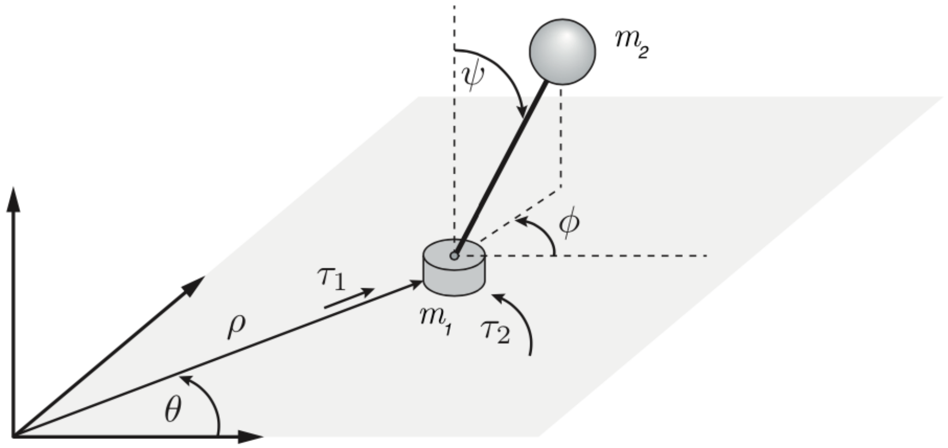

Figure 4.2: Schematic diagram of spherical pendulum system [2, 5]

(1) The smooth simple simple mechanical control system $(Q, \mathbb{G}, F, P)$ is defined by:

- **Configuration manifold** $Q = \mathbb{R}_{>0} \times \mathbb{S}^1 \times \mathbb{S}^2$, with coordinates $q = (\rho, \theta, \phi, \psi)$.
- **Kinetic metric** $\mathbb{G} = \frac{1}{2} m_1 \dot{x}_1^2 + \frac{1}{2} m_2 \dot{x}_2^2$
- **Actuation** $F = (\mathrm{d}\rho, \mathrm{d}\theta)$
- **Potential** $P = -g m_2 z = -g m_2 (\ell \cos \psi)$

From the diagram of the system, it can be shown that the spatial coordinates are

$$\vec{x}_1 = \begin{pmatrix} \rho \sin \theta \\ \rho \cos \theta \\ 0 \end{pmatrix}, \tag{4.17}$$

$$\vec{x}_2 = \vec{x}_1 + \begin{pmatrix} \sin \psi \cos \phi \\ \sin \psi \sin \phi \\ \cos \psi \end{pmatrix}. \tag{4.18}$$

The corresponding velocities are given by,

$$\dot{\vec{x}}_1 = \begin{pmatrix} \rho \dot{\theta} \cos \theta + \dot{\rho} \sin \theta \\ -\rho \dot{\theta} \sin \theta + \dot{\rho} \cos \theta \\ 0 \end{pmatrix}, \tag{4.19}$$

$$\dot{\vec{x}}_2 = \dot{\vec{x}}_1 + \ell \begin{pmatrix} \dot{\psi} \cos \psi \cos \phi - \dot{\phi} \sin \psi \sin \phi \\ \dot{\psi} \cos \psi \sin \phi + \dot{\phi} \sin \psi \cos \phi \\ -\dot{\psi} \sin \psi \end{pmatrix}. \tag{4.20}$$

(2) The system's Lagrangian depends on the kinetic energies and potentials of each point mass.

The kinetic energy of mass m_1 is:

$$G_1(\dot{q}) = \frac{1}{2}m_1\dot{x}_1^2 = \frac{1}{2}m_1\left(\rho^2\dot{\theta}^2 + \dot{\rho}^2\right). \tag{4.21}$$

The kinetic energy of mass m_2 works out to be:

$$\begin{aligned} G_2(\dot{q}) &= \frac{1}{2}m_2\dot{x}_2^2 && (4.22)\\ &= \frac{1}{2}m_2\bigg[\left(\rho^2\dot{\theta}^2 + \dot{\rho}^2\right)\\ &\qquad + \rho\ell\dot{\theta}\left(\dot{\psi}\cos\theta\cos\psi\cos\phi - \dot{\phi}\cos\theta\sin\psi\sin\phi - \dot{\psi}\sin\theta\cos\psi\sin\phi - \dot{\phi}\sin\theta\sin\psi\cos\phi\right)\\ &\qquad + \ell^2\left(\dot{\psi}^2 + \sin^2\psi\dot{\phi}^2\right)\bigg] && (4.23)\end{aligned}$$

Finally, the potential energy is:

$$P(q) = -gm_2\ell\cos\psi \tag{4.24}$$

Finally, we put the pieces together to obtain the Lagrangian of the system. The simplified result is given by,

$$\begin{aligned} \mathcal{L}(q,\dot{q}) = \frac{1}{2}&(m_1+m_2)\left(\rho^2\dot{\theta}^2 + \dot{\rho}^2\right)\\ &+ m_2\rho\ell\dot{\theta}\left(\dot{\psi}\cos\psi\cos(\theta+\phi) - \dot{\phi}\sin\psi\sin(\theta+\phi)\right)\\ &+ \frac{1}{2}m_2\ell^2\left(\dot{\psi}^2 + \sin^2\psi\dot{\phi}^2\right)\\ &- gm_2\ell\cos\psi.\end{aligned} \tag{4.25}$$

(3) From Equation 4.25, the system's symmetry is found to be in the sum $\xi := \theta + \phi$. Define the left action of the Lie group $\mathbb{S}^1$ to be $\Phi : Q \times \mathbb{S}^1 \to Q$ to be the symmetry of the system, where the configuration space is $Q = \mathbb{R}^+ \times \mathbb{S}^1 \times \mathbb{S}^1$.

$$\Phi_\pm : \begin{pmatrix}\rho\\ \theta\\ \phi\\ \psi\end{pmatrix} \in Q, g \in G \mapsto \begin{pmatrix}\rho\\ \theta \pm g\\ \phi \mp g\\ \psi\end{pmatrix} \in Q. \tag{4.26}$$

We can see that for all $g \in G$, this symmetry has no effect on ξ:

$$\xi(q) := \theta + \phi, \tag{4.27}$$

$$\xi\left(\Phi_\pm(q,g)\right) = (\theta \pm g) + (\phi \mp g) = \theta + \phi \pm g \mp g = \xi(q). \checkmark \tag{4.28}$$

Since the Lagrangian's θ, ϕ dependency can be expressed as a function of ξ, the Lagrangian is also invariant under the action of Φ.

(4) McCarthy chose the following VHCs for this problem,

$$h_1(q) := \theta - \psi, \tag{4.29}$$

$$h_2(q) := \psi - \frac{\pi}{4}\tanh\rho, \tag{4.30}$$

from which we get

$$\mathrm{d}h_q = \begin{pmatrix} 0 & 1 & -1 & 0 \\ \frac{\pi}{4}\tanh^2(\rho) - \frac{\pi}{4} & 0 & 0 & 1 \end{pmatrix}, \tag{4.31}$$

$$\mathrm{Ker}\left(\mathrm{d}h_q\right) = \left\{ \begin{pmatrix} 0 \\ 1 \\ -1 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \\ 0 \\ -\frac{\pi}{4}\tanh^2(\rho) + \frac{\pi}{4} \end{pmatrix} \right\}. \tag{4.32}$$

The set of chosen VHCs $h(q)$ (Definition 2.1.3) constrain the system to a submanifold $\mathcal{C} \subset Q$, with

$$T_q\mathcal{C} := \mathrm{Ker}\left(\mathrm{d}h_q\right). \tag{4.33}$$

This tangent space can equivalently be defined using the Ehresmann connection.

$$T_q\mathcal{C} = \mathrm{Span}\left(V(q), H(q)\right). \tag{4.34}$$

Take $V(q)$ as the Vertical Vector Field defined by the direction of the symmetry:

$$V(q) := \frac{\partial \Phi_\pm}{\partial g} = \frac{\partial}{\partial g}\begin{pmatrix} \rho \\ \theta \pm g \\ \phi \mp g \\ \psi \end{pmatrix} = \begin{pmatrix} 0 \\ \pm 1 \\ \mp 1 \\ 0 \end{pmatrix} \overset{\text{WLOG}}{=} \begin{pmatrix} 0 \\ 1 \\ -1 \\ 0 \end{pmatrix}. \tag{4.35}$$

The robot arm system has 4 degrees of freedom and 2 actuators, equating to $4 - 2 = 2$ degrees of underactuation. Since we plan to use this actuator to enforce the VHC, the constraint manifold should have $\dim \mathcal{C} = 2$ remaining degrees of freedom.

Since $V(q)$ in Equation 4.35 also appears in the basis for $T_q\mathcal{C}$ (see Equation 4.32), it is natural to choose the horizontal component as the second element

$$H(q) = \begin{pmatrix} 1 \\ 0 \\ 0 \\ -\frac{\pi}{4}\tanh^2(\rho) + \frac{\pi}{4} \end{pmatrix}, \tag{4.36}$$

and their direct sum defines the fiber bundle $T\mathcal{C} = TV \oplus TH$.

(5) The constrained dynamics are computed using the expression with constrained Christoffel Symbols in Equation 3.9.

The Christoffel Symbols (Definition 2.3.2) of the constraint manifold $\overset{\mathcal{C}}{\Gamma}{}^k_{ij}$ are computed in Matlab.

CHAPTER 5

Discussion

5.1 Summary of Findings

5.1.1 Robotic Arm Offset

The simulation was able to show convergence of the VHCs for a large range of initial conditions. Figure 5.1 shows a few sample plots. Note that the VHC was adjusted to account for the behaviour of angles as $\mathbb{R}/2\pi$.

One plot, in the top left, appears as not converging: this corresponds to an initial where $\theta = \phi$. In fact, this is just a floating point error and the divergence is on the order of $10^{-}13$. While not a critical issue, it is recommended that future iterations of the code account for these arithmetic inconsistencies.

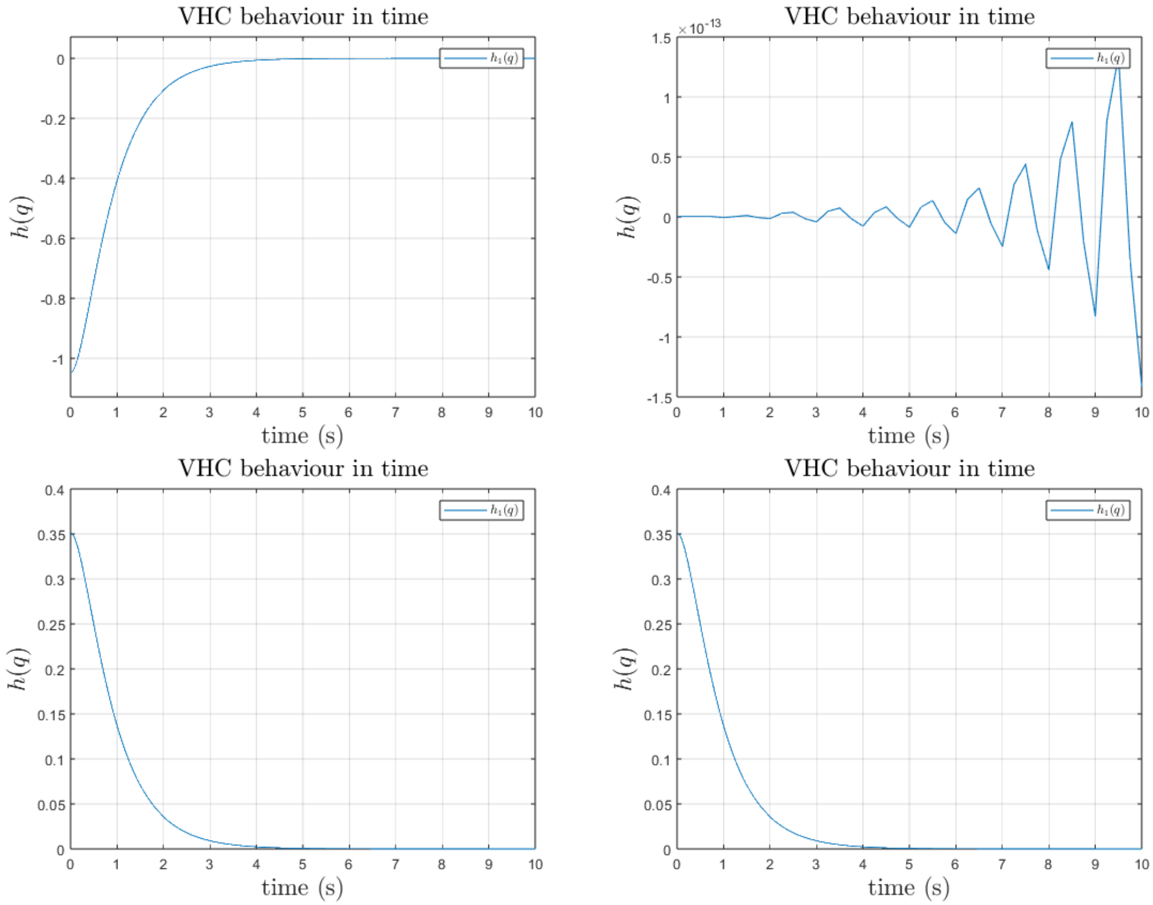

Figure 5.1: Plot showing convergence of the VHCs (control error) towards zero on the Offset Robotic Arm model. Here, the constraint is defined using the equivalent $h := \arg\left\{\exp\left(i[\theta - \phi]\right)\right\}$.

5.1.2 Spherical Pendulum

Figure 5.2 shows convergence for the spherical pendulum subject to a range of initial conditions. The top left plot appears to converge slowly, and times out before the VHC gets close to zero. In this case, it is recommended to review the control constants K_P, K_D to allow for faster convergence. The LQR function in Matlab can also be used to achieve optimal control.

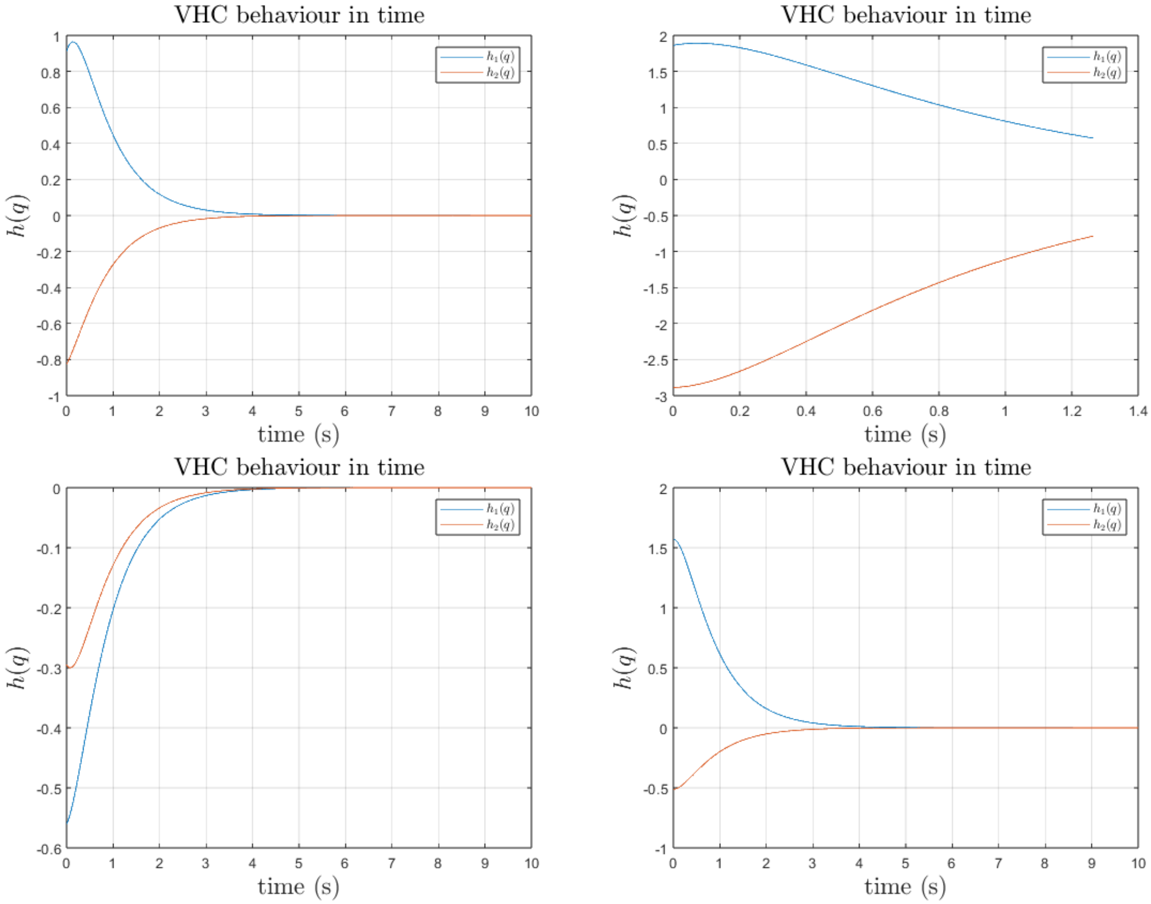

Figure 5.2: Plot showing convergence of the VHCs (control error) towards zero on the Spherical Pendulum model. Here, the constraint is defined $h_1 := \theta - \phi, h_2 = \psi - \frac{\pi}{4} \tanh \rho$.

5.2 Future Research

5.2.1 Analyze further examples

From Section 4.1, the Robotic Arm Offset example is somewhat trivial, as its reduced form is apparent at all points on the constraint manifold $\mathcal{C}$.

Therefore, future research should explore further examples of mechanical systems, with the aim to provide a more concrete illustration of the technique of dynamical reduction described by McCarthy et al[2]. Tedrake's notes on Underactuated Robotics[3] contain some examples that (with perhaps some modifications) could be used in a continuation of this work. It is recommended to choose systems with 4 degrees of freedom and

2 actuators–systems with 3 degrees of freedom and 1 actuator may not be as interesting to study.

5.2.2 Metrizability of the connection

If the connection on the constraint manifold is metrizable, the dynamics of the VHC-constrained system are Euler-Lagrange[2]. It would be of interest to determine whether this applies to both systems. Euler-Lagrange systems are well-studied, and if the dynamics are indeed so, the system would be easy to model using the tried and tested strategies.

Chapter 6

Conclusion

In conclusion, this project demonstrated the use of Virtual Holonomic Constraints in simple, 2$^{\text{nd}}$-degree underactuated mechanical systems with 1 degree of symmetry. Using Matlab, the constrained dynamics of these systems were analyzed and simulated. The systems' constrain manifolds were decomposed into an Ehresmann-like connection of vertical and horizontal vector fields. This dynamical reduction technique can be useful in simplifying the dynamics of robotic applications that require the enforcement of Virtual Holonomic Constraints, such as maintaining set distances between agents or synchronizing the motions of joints. Overall, the study of virtual holonomic constraints has potential applications in the field of robotics and further research in this area may lead to new and improved control strategies.

Acknowledgements

To my supervisor, Professor Chris Nielsen: a sincere thank you for taking me as a student this term. While we had to navigate a few logistical obstacles before successfully launching the project, things went well as we progressed. I learned a lot about geometric mechanics, controls, and about the research process. I'm thankful for your mentorship, advice, and patience along the way.

Thank you to my academic advisor, Professor Richard Epp, for encouraging me to take on a project, and for supporting me through the final year of my degree.

Thank you to my friends Michael Astwood and Basel Jayyusi for teaching me some differential geometry fundamentals, and reminding me of the beauty of mathematics.

Bibliography

[1] WikimediaCommons, “Gsma mobile world congress 2021,” 2021. https://commons.wikimedia.org/wiki/File:MWC21_-_25.jpg.

[2] P. J. McCarthy, M. Maggiore, C. Nielsen, and L. Consolini, “A noether theorem and symmetry reduction for forced affine connection systems on 2-manifolds.” Preprint, 2020.

[3] R. Tedrake, *Underactuated Robotics: Algorithms for Walking, Running, Swimming, Flying, and Manipulation*. 2022. Course Notes for MIT 6.832.

[4] F. Bullo and A. D. Lewis, *Geometric control of mechanical systems: modeling, analysis, and design for simple mechanical control systems*, vol. 49. Springer, 2019.

[5] M. Maggiore and L. Consolini, “Virtual holonomic constraints for euler–lagrange systems,” *IEEE Transactions on Automatic Control*, vol. 58, no. 4, pp. 1001–1008, 2012.

[6] G. F. Franklin, J. D. Powell, A. Emami-Naeini, and J. D. Powell, *Feedback control of dynamic systems*, vol. 4. Prentice hall Upper Saddle River, 2002.

[7] C. Nielsen, “Note - jan. 19, 2021.” Personal notes, 2021.

[8] N. A. F. Senan, “A brief introduction to using ode45 in matlab,” *University of California at Berkeley, USA*, 2007.

[9] J. E. Marsden and T. S. Ratiu, *Introduction to mechanics and symmetry: a basic exposition of classical mechanical systems*, vol. 17. Springer Science & Business Media, 2013.

[10] L. Consolini, M. Maggiore, C. Nielsen, and M. Tosques, “Path following for the pvtol aircraft,” *Automatica*, vol. 46, no. 8, pp. 1284–1296, 2010.

[11] L. Consolini and M. Maggiore, “Control of a bicycle using virtual holonomic constraints,” *Automatica*, vol. 49, no. 9, pp. 2831–2839, 2013.

[12] J. Oprea, *Differential geometry and its applications*, vol. 59. American Mathematical Soc., 2019.

[13] J. Baxt, “File:cpccg screening robot.jpg,” 2021. https://upload.wikimedia.org/wikipedia/en/thumb/2/27/CPCCG_screening_robot.jpg/1600px-CPCCG_screening_robot.jpg?20090730203425.

[14] O. S. U. Anna Davis, “pendulumswing.jpg,” 2013. https://ximera.osu.edu/ode/main/simplePendulum/simplePendulum.

[15] D. T. Tips.