



Fachbereich 4 Produktionstechnik
Masterstudiengang Systems Engineering

Enabling Autonomous Robot Locomotion through Self-Explored Sensorimotor Manifolds

(Autonome Fortbewegung eines Roboters auf Basis seiner
selbsterkundeten sensomotorischen Mannigfaltigkeiten)

Masterarbeit zur Erlangung des akademischen Grades
Master of Science (M. Sc.)

Gutachter

Prof. Dr. Udo Frese, Universität Bremen

Prof. Dr. Manfred Hild, Berliner Hochschule für Technik

Eingereicht von

Vincent Rist

Martikeldnummer: 6216470

E-Mail: rist@uni-bremen.de

Bremen, 28. Juli 2025

Abstract

Autonomous robots should be able to explore information about themselves and their environment to move meaningfully within it. One promising approach is the Attractor-based Behavior Control (ABC), an intrinsically motivated exploration strategy that discovers intuitive, energy-efficient postures. These postures correspond to equilibria of the underlying dynamical system, connected by heteroclinic orbits. The resulting topological structure can be stored as a graph that represents a compact discretization of the sensorimotor manifold. During deployment on the physical robot, nonlinearities in the motor control, such as dead-zones, can lead to uncertainties in the exploration graph. In this work, I propose an extension of the ABC method that exploits the gravity-induced asymmetry of the dead-zone for reliable convergence. I then suggest a methodology to infer a locomotion mode from the topological structure of the exploration graph. A physical, planar robot serves as an experimental platform for the development and evaluation of the presented algorithms.

Kurzfassung

Autonome Roboter sollten in der Lage sein, Informationen über sich selbst und ihre Umgebung zu erkunden, um sich sinnvoll darin zu bewegen. Ein vielversprechender Ansatz ist das Attractor-based Behavior Control (ABC), eine intrinsisch motivierte Erkundungsstrategie, die intuitive, energieeffiziente Körperhaltungen entdeckt. Diese Haltungen entsprechen Ruhelagen des zugrunde liegenden dynamischen Systems, die durch heterokline Orbits miteinander verbunden sind. Die resultierende topologische Struktur kann als Graph gespeichert werden, der eine kompakte Diskretisierung der sensomotorischen Mannigfaltigkeit darstellt. Während der Inbetriebnahme am physischen Roboter können Nichtlinearitäten in der Motorsteuerung wie Totzonen zu Unsicherheiten im Explorationsgraphen führen. In dieser Arbeit schlage ich eine Erweiterung des ABC-Verfahrens vor, die die schwerkraftbedingte Asymmetrie der Totzone für eine zuverlässige Konvergenz ausnutzt. Anschließend schlage ich eine Methodik vor, um aus der topologischen Struktur des Explorationsgraphen auf eine Fortbewegungsart zu schließen. Ein physischer, planarer Roboter dient als experimentelle Plattform zur Entwicklung und Bewertung der vorgestellten Algorithmen.

Contents

1	Introduction	1
1.1	Background and Motivation	1
1.2	Problem Statement	3
1.3	Outline	4
2	Related Work	5
2.1	Locomotion in Robotics	5
2.2	Dead-zone Compensation	5
2.3	Sensorimotor Manifolds	6
2.4	Exploration and ABC-related Research	7
3	Theoretical Foundations	9
3.1	Topological Terminology	9
3.2	Dynamical Systems	12
3.2.1	Stability	14
3.2.2	Attractors	16
3.2.3	Bifurcations	18
3.3	Sensorimotor Systems	20
3.3.1	Gravity	20
3.3.2	Sensorimotor Manifolds	24
3.3.3	Cognitive Sensorimotor Loop (CSL)	28
3.4	Attractor-based Behavior Control	30
3.4.1	A Toy Example	31
3.4.2	Heteroclinic Orbits	32
3.4.3	Breaking the Symmetry	33
3.4.4	ABC Graph Structure	34
3.4.5	Stall	35
4	Robotic Platform	37
4.1	Hardware and Electronics	37
4.2	Morphology	40
4.3	Simulation Framework	43

5	Self-Exploration	44
5.1	Equilibrium Detection	45
5.1.1	Fixpoint Detection	45
5.1.2	Oscillation Detection	48
5.2	Dead-zone	50
5.2.1	Formal Definition	50
5.2.2	Dead-zone-induced Uncertainty	53
5.2.3	Dead-zone Asymmetry	53
5.2.4	Dead-zone Control	57
5.3	Exploration Results	59
6	Locomotion	61
6.1	Sliding-style Locomotion	61
6.2	Topologically Inferring Locomotion	62
6.3	Planning	65
6.4	Locomotion Results	66
7	Conclusion	69
7.1	Summary	69
7.2	Limitations	70
7.3	Outlook	71
	List of Figures	72
	List of Tables	73
	List of Algorithms	73
	References	74

List of Abbreviations

Abbreviation	Meaning
x D	x -dimensional
AB6D	AccelBoard6D
ABC	Attractor-based Behavior Control
AE	Artificial Evolution
AI	Artificial Intelligence
COM	Center of Mass
CSL	Cognitive Sensorimotor Loop
DOF	Degree of Freedom
DZC	Dead-zone Control
FPD	Fixpoint Detector
GCP	Ground Contact Point(s)
GNG	Growing Neural Gas
IMU	Inertia Measurement Unit
IIR	Infinite Impulse Response
NASA	National Aeronautics and Space Administration
OCD	Oscillation Detector
PID	Proportional-Integral-Differential (controller)
PLA	Poly lactide Acid
PWM	Pulse Width Modulation
RL	Reinforcement Learning
SOM	Self-Organizing Map
SRAM	Static Random Access Memory
SMM	Sensorimotor Manifold
STL	Standard Tessellation Language (3D file extension)

List of Symbols

Notation	Description
<i>Mathematical Prerequisites</i>	
$\mathbb{Z}$	Set of whole numbers
$\mathbb{R}$	Set of real numbers
$\mathbb{C}$	Set of complex numbers
$\text{Re}: \mathbb{C} \rightarrow \mathbb{R}$	Real part of a complex number
$\text{Im}: \mathbb{C} \rightarrow \mathbb{R}$	Imaginary part of a complex number
$f \circ g = f(g)$	Function composition
$f^n = \underbrace{f \circ f \cdots f}_{n \text{ times}}$	
$\mathbf{x} = [x_1, x_2, \cdots, x_n]^T$	Column vector
$\mathbf{0} = [0, 0, \cdots]^T$	Zero vector
$x_i = \mathbf{x}[i]$	Vector components (scalar)
$\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_m]$	Matrix of column vectors
$\lambda_i, \mathbf{v}_i$	Eigenvalue, eigenvector
$\mathbf{X}\mathbf{v}_i = \lambda_i\mathbf{v}_i$	Eigendecomposition
∇f	Nabla-operator, gradient of a function
$\mathbf{J}_f$	Jacobian matrix of a function
$\approx$	Approximation
$\propto$	Proportionality
<i>Set Notation</i>	
$\{\}$	Set construction
$\in$	Membership
$\exists$	Existence
$\subset$	Subset
$\setminus$	Set difference
$\cup, \bigcup$	Set union
<i>Topological Terminology</i>	
$\mathbb{R}^n$	Euclidean space
$[x, y, z, \cdots] = \mathbf{x}$	Cartesian coordinates
$\mathbb{G} = (V, E)$	Graph
$\mathbb{M}$	Manifold
$\mathbb{S}^n$	n -sphere

$\mathbb{T}^n$	n -torus
$\mathbb{A} = [-\pi, \pi)$	Angular space
<i>Dynamical Systems</i>	
$t \in T$	Time
$T = \mathbb{R}$	Continuous time domain
$T = \mathbb{Z}$	Discrete time domain
$\mathbf{s}(t) \in S$	State at time t
$\mathbf{s}(0)$	Initial state
$\mathbf{s}(a : b) = \{\mathbf{s}(t)\}_{t \in [a, b)}$	Trajectory
$\Phi : S \times T \rightarrow S$	Flow
$\dot{\mathbf{s}}(t) = \frac{d}{dt}\mathbf{s}(t)$	State derivative, continuous velocity
$f(\mathbf{s}(t)) = \dot{\mathbf{s}}(t)$	Vector field, continuous DS
$\Delta \mathbf{s}(t) = \mathbf{s}(t) - \mathbf{s}(t-1)$	State difference, discrete velocity
$F(\mathbf{s}(t)) = \mathbf{s}(t+1)$	Iterative map, discrete
$\mathbf{s}_*$	Fixpoint, equilibrium
$\mathbf{s}_\bullet, \bullet$	Stable fixpoint
$\mathbf{s}_\circ, \circ$	Unstable fixpoint
$\mathbf{s}_\bullet, \bullet$	Saddle point (s_0 is stable)
$\mathbf{s}_\circ, \circ$	Saddle point (s_1 is stable)
$A(\mathbf{s}) \subset S$	Attractor of an initial state
$B(A_i) \subset S$	Basin of an attractor
$\mathcal{A}$	Set of all attractors
$\mathcal{B}(\mathbf{s})$	Bifurcation set
<i>Robotic System</i>	
φ	Orientation, body angle in world frame
θ	Robot configuration parameters
$\theta_L, \circlearrowleft$	Left leg joint angle, marker
$\theta_R, \circlearrowright$	Right leg joint angle, marker
$\mathbf{c} = [c_x, c_y], \odot, \odot$	Center of mass (COM)
$\mathbf{p}_i = [p_{xi}, p_{yi}], \blacktriangle$	i -th ground contact point (GCP)
u	Voltage [V]
i	Current [A]
$p = ui$	Power [W]
<i>ABC Notation</i>	
g_i	CSL input gain

g_f	CSL feedback gain
r	Release-mode
c	Contraction-mode
$m \in \{r, c\}$	CSL mode
$\mathbf{m} = (m_L, m_R)$	2D state mode
$+$	active switch in positive direction
$-$	active switch in negative direction
$\pm$	active switch (either direction)
$\cdot$	passive, no switch
$d \in \{+, -, \cdot\}$	switch direction
$\mathbf{d} = (d_L, d_R)$	2D switch mode
<i>Heteroclinic orbits</i>	
$\mathbf{s}_{*i} \xrightarrow{\mathbf{d}} \mathbf{s}_{*j}$	unidirectional
$\mathbf{s}_{*i} \xleftrightarrow{\mathbf{d}} \mathbf{s}_{*j}$	bidirectional
$\mathbf{s}_{*i} \circlearrowright^{\mathbf{d}}$	self-referencing loop, stall connection
<i>Algorithm Notation</i>	
FUNCTION()	Function
$=$	Variable definition
$\leftarrow$	Value assignment
<i>Miscellaneous</i>	
f_s	Sampling frequency
$\alpha \in [0, 1]$	Factor for exponential low-pass filter
$\tilde{\theta}$	Smoothed sensor value
(d, r_1, r_2, γ)	Cashew arc shape parameters
$D(u)$	Dead-zone function
Δp	Distance between GCP
w	Normalized weight distribution
$\Delta\theta_{(\cdot)}$	passive angular displacement

1 Introduction

“Yes,” said Sir. “That’s exactly what I do think, Andrew. Well beyond your programmed levels, as a matter of fact. Not that I’m troubled that you’ve unexpectedly turned out to have this streak of artistic ability in you, you understand. But I’d like to know just why it’s there.”

(Isaac Asimov in *The Positronic Mac*, End of Chapter 3, [1])

With this exchange between Sir and his humanoid robot Andrew, who unexpectedly—beyond his original programming—develops wood-carving skills, Isaac Asimov captures a fundamental motivation in robotics: *emergence*, the concept where complex behaviors arise from simple rules. Asimov’s Three Laws of Robotics state:

1. A robot may not injure a human being, or, through inaction, allow a human to come to harm.
2. A robot must obey the orders given it by human beings except where such orders would conflict with the First Law.
3. A robot must protect his own existence as long as such protection does not conflict with the First or Second Law.

Sir’s daughter instructs Andrew to carve a gift for her out of a piece of wood. This order together with the Second Law, forces Andrew to become creative and acquire a new skill beyond his intended programming. From this story, we can see how this ability is not only useful but necessary if autonomous robots are ever to really be involved in our everyday lives. Beyond some constrained specific cases like vacuuming or entertainment, Asimov’s fantasy remains largely fictional.

1.1 Background and Motivation

The pursuit of autonomy is of central interest to the field of robotics. Autonomy can be understood as self-reliance, independence, or freedom of choice. A robot remotely controlled by humans has little to no autonomy. Industrial robots that automate a programmed process move autonomously but do not

make their own decisions. Adaptive learning processes, on the other hand, can adjust to different conditions and react to disturbances, but they too were created by a human. The ultimate aspiration is—as articulated by Rodney Brooks in his famous paper *Intelligence without Representation*[2]:

to build completely autonomous mobile agents that co-exist in the world with humans, and are seen by those humans as intelligent beings in their own right.

Brooks therefore popularized the concept of *embodiment*, according to which true intelligence emerges from an agent situated in a physical environment. The term *agent* refers to an abstract concept of a brain or controller (the software) and a body or morphology (the hardware) that serves as an interface to the environment. An artificial mechanical agent is what we call a *robot*. The agent and the environment communicate via two channels: the agent outputs *motor* signals to act upon the environment and receives *sensor* signals as input to observe it. These data streams are usually highly interconnected and form complex topological structures called *sensorimotor manifolds*. These manifolds, though challenging to interpret for humans, are implicit body models unique to the morphology and environment which is why they are a promising representation for behavior planning.

Without computationally expensive onboard physics engines, autonomous robots need to discover these complex manifolds themselves upon deployment. This process is called *exploration*. The prefix in *self*-exploration serves a dual purpose: on one hand, it emphasizes that exploration includes—beyond a spatial mapping of the environment—the discovery of one’s own body. On the other hand, it refers to the *motivation*, the reason an agent explores in the first place. The problem of how to best motivate a robot to engage with and learn about the world is called the *lazy robot problem*. Empirical methods like reinforcement learning (RL) or artificial evolution (AE) motivate their agents with a reward or a fitness function that needs to be optimized. But this function is still designed by human hands and can therefore not really be labeled as “self”-exploration.

This is where the Attractor-based Behavior Control (ABC) comes in. It proposes an exploration paradigm that is self- or *intrinsically* motivated and operates without a human-designed fitness function. The ABC views the

agent-environment relationship through the lens of *dynamical systems*, a mathematical framework that studies and classifies the convergence behavior of systems. An ABC-based exploration run on a robot naturally discovers the system’s *states of equilibrium* that correspond to energy-efficient body postures. With *postures*, I refer to specific positioning of the robot’s limbs. Adjacent postures are connected via motor movements which materialize a network structure or *graph*. This graph is a sparse discretization of the sensorimotor manifold and therefore an ideal tool for graph-based behavior planning.

One essential behavior of autonomous robots is *locomotion*: the ability to deliberately move from one place to another. Not only do many locomotion techniques exist, i. e., running, rolling, or jumping, but even the same techniques can vary strongly depending on the morphology. Ideally, autonomous robots should have the ability to come up with a locomotion style that suits their morphology and environment without human help.

1.2 Problem Statement

This work investigates a *planar* robotic platform, where the morphology is designed in the two-dimensional sagittal plane and extruded linearly in the third dimension. Motor axes are oriented orthogonally to this plane to maintain movement constraints. The robot comprises three limbs with two actuated joints, providing a simplified yet physically embodied system as a playground for developing and testing fundamental concepts.

The central research question addressed is:

Can a locomotion behavior for a planar, three-limb robot be derived solely from topological analysis of his exploration graph?

To answer this question, I establish two primary objectives:

1. to develop a practical implementation of ABC that accounts for motor nonlinearities, particularly dead-zone effects that introduce uncertainty in state transitions, and
2. to formalize a motion generation algorithm that abstracts behavioral patterns from morphological properties through geometric and topological graph analysis.

1.3 Outline

Section 2 (Related Work) surveys foundational research and establishes the theoretical context for this contribution.

Section 3 (Theoretical Foundations) formally introduces the dynamical systems framework and mathematical notation underlying the approach.

In Section 4 (Robotic Platform), I present the robot *Brilliance*, detailing its hardware implementation, morphological design, and simulation environment, with particular attention to the design rationale.

Section 5 (Exploration) demonstrates the practical implementation of the Attractor-based Behavior Control, including methods for equilibrium detection and dead-zone compensation.

Section 6 (Locomotion) develops a topology-driven motion planning methodology, showing how morphological properties can be inferred from exploration graphs to generate functional behaviors.

Section 7 concludes this thesis by providing a synthesis of the contributions, discussion of limitations, and directions for future research.

2 Related Work

The study of autonomous robots has led to significant advances which are summarized here into four main subsections that are relevant for this thesis: firstly, an overview of locomotion in robotics; secondly, existing compensation techniques for dead-zones and other common nonlinearities; thirdly, research on and representation of sensorimotor manifolds; and finally, exploration with particular focus on previous work of the ABC approach.

2.1 Locomotion in Robotics

“Locomotion, the ability of a body to move from one place to another.”[3] The periodic motion sequence that enables an agent to locomote is called *gait*. Naturally, a gait is highly dependent on the physical environment and the agent’s morphology. Hence an incredibly diverse set of gaits exists: swimming underwater, flying in the air, walking on land, etc.

Even on flat, rigid surfaces, endless locomotion techniques can be found: crawling, rolling, slithering, jumping, walking, running, etc. Some animals, like horses, can even vary between different gaits to achieve greater speeds or lower energy expenses[4]. Many robots have been specifically designed to embody a certain type of gait, be it quadrupeds[5], hexapods[6], monopod hoppers[7], powered bipeds[8], hyper-redundant serpentine robots[9], soft robotics[10] or even modular reconfigurable robots[11].

McGeer introduced the idea of Passive Dynamic Walkers[12], a bipedal contraption that could walk down a slope entirely without control. Based on this, hybrid models have been designed that try to generate efficient gaits with minimal actuation to replace gravity as an energy source[13].

2.2 Dead-zone Compensation

Jelaili and Huang define the *dead-zone* as a “predetermined range of input through which the output remains unchanged irrespective of the direction of change”[14]. Robotic systems rely on the use of electrical motors to navigate in the world and achieve tasks like locomotion. Even though the output velocity is usually linearly coupled with the input voltage of the motor, most physical motors suffer from a dead-zone around an input of zero. A critical threshold

must be reached before the output reacts. This phenomenon is typically attributed to friction[15], the shearing force that opposes the movement of two surfaces in contact. Friction can be reduced through good hardware design or the use of lubrication but not fully removed.

Several control methods use parametrized periodic signals superimposed onto the primary control signal. A high-frequency vibration-like signal can create an effective narrowing of the nonlinear sector[16]. Square pulsed excitations have successfully reduced the control error during stick-slip motion of pneumatic valves[17]. Though these methods are often easily implemented, they introduce wear and tear.

Adaptive control theory offers numerous established approaches that learn dead-zone parameters online and compensate through an inverse function[18]. Joint friction can also be adaptively compensated by a velocity observer to achieve higher control accuracy[19].

Moreover, empirical approaches show promising results for irregular dead-zone profiles where an exact model is unavailable. Fuzzy pre-compensators obtained from evolutionary programming have been shown to improve the performance of PD-controllers[20]. Selmic et al. have effectively demonstrated that neural network-based approaches can successfully compensate for dead-zones[21] and other common nonlinearities like backlash[22] or saturation[23].

2.3 Sensorimotor Manifolds

When making sense of complex data, the need to model and predict the topological nature of it becomes necessary. Early approaches include Kohonen’s self-organizing maps (SOM)[24], Kuipers cognitive map[25], or Fritzke’s growing neural gas (GNG)[26]. Expanding these ideas to sensorimotor data leads to the sensorimotor map[27]: a neural representation on which to realize planning.

Experiments with NASA’s humanoid robot *Robonaut* have shown that dimensionality reduction methods can be used to reveal the topological nature of high-dimensional sensorimotor data streams[28]. The lower-dimensional representations allow extraction of behavioral manifolds related to a humanoid robot’s body postures[29].

Continuous representations of sensorimotor manifolds have been developed. For example, *Lie groups*: smooth manifolds endowed with a group struc-

ture, allow geometric interpretation of motor capabilities for mobile robots[30]. Custom neurons can be used to represent *quadrics*—a manifold defined by a second-degree polynomial—and learn its parameters to approximate a set of data points[31].

SMM representations have been used to plan behavior and accomplish tasks. The humanoid robot Myon could accomplish different visuomotor tasks by generating and optimizing quadric-based representations[32]. Gaits synthesized via graph-based representations of the sensorimotor manifold have enabled 3D-printed planar robots to locomote[33].

2.4 Exploration and ABC-related Research

Benureau formalizes exploration as “creating access to different aspects of the environment”[34]. This incorporates the need to create spatial maps and localization within them[35] but can also be generalized to the discovery of the law-like sensorimotor contingencies[36]. Autonomous systems have no prior knowledge about themselves or the environment; hence, exploration becomes a necessary predecessor to the planning of behavior.

Empirical approaches are based on random trial and error to determine statistical models like in motor babbling[37] or reinforcement learning[38]. But these approaches heavily suffer from the curse of dimensionality: the number of possible inputs grows exponentially with increasing complexity. The therefore large number of necessary trials to adequately sample the state-action space means high computational intensity for simulations or time, wear, and cost intensity for physical systems.

To address this, Hild introduced the Attractor-based Behavior Control (ABC), an exploration algorithm that creates a graph of intuitive postures:

Interestingly, people immediately name the postures found by ABC-Learning with verbs like lying, sitting, standing, kneeling, holding up, which seems plausible if we realize that, e. g., the verb sitting represents all variations of sitting, be it on a chair with the feet on the ground, or on a table with dangling feet, but always the defining property being that hip and neck are held upright (contraction mode), whereas knees and ankles are free to move (release mode).[39]

The ABC graph is a sparse discretization of the SMM whose complexity only grows linearly with increasing degrees of freedom (DOF). It therefore presents a promising approach to taming the curse of dimensionality.

The ABC employs Cognitive Sensorimotor Loops (CSL)[40] which naturally locate stable fixpoints when driven in *release*-mode and unstable fixpoints when driven in *contraction*-mode. The CSL is a simple sensorimotor coupling from which surprisingly complex behavior emerges[41]: A robot leg with three joints controlled by independent contraction-mode CSLs moves from a horizontal lying pose into an upright standing pose without any trajectory planning[42].

Extensions to the original ABC concept have been proposed. Werner studied the *kick-fly-catch* paradigm that uses initial energy impulses and momentum to improve energy efficiency of contraction-mode transitions[43]. Bethge evaluated different heuristics-based exploration strategies that improve the exhaustiveness or efficiency of the ABC algorithm[44]. Janz has pioneered the idea of comparing past exploration data to current data to recognize situations and to classify obstacles through deviations from expected behavior[45].

3 Theoretical Foundations

This section presents the theoretical and mathematical foundations underpinning the concepts developed in this thesis. Firstly, key terminology from the field of topology is introduced. Next, the mathematical formulation of dynamical systems is provided, establishing the basis for a concrete example system involving a robot morphology in a gravity-driven environment. Finally, the section describes the Cognitive Sensorimotor Loop (CSL) as the foundational component of the Attractor-based Behavior Control (ABC), a framework enabling robots to self-explore their sensorimotor manifolds.

3.1 Topological Terminology

This thesis incorporates and extends key concepts from the mathematical field of topology that are abstractly introduced in this subsection.

Euclidean Space, denoted as $\mathbb{R}^n$, serves as the geometric model of physical space. In this space, any point can be uniquely represented by coordinates relative to a chosen origin. n corresponds to the number of dimensions required for its description: $n = 1$ given an infinite line, $n = 2$ represents an infinite plane, and so forth. The coordinates of a point or vector in the $\mathbb{R}^n$ are conventionally expressed as:

$$[x, y, z, \dots] = \mathbf{x} \in \mathbb{R}^n$$

Euclidean space inherently defines fundamental geometric properties, including distance, angle, and direction. The distance between a point and the origin is modeled with the Euclidean distance:

$$\|\mathbf{x}\| = \sqrt{x^2 + y^2 + z^2 + \dots}.$$

Topological Spaces represent a generalization of Euclidean spaces. Unlike a Euclidean space, a topological space does not necessarily require a notion of distance between its elements. Instead, it characterizes the spatial structure of its elements in terms of their relative position and ordering. Several examples of topological spaces are presented subsequently.

Homeomorphism refers to a bijective mapping between two spaces that preserves all topological properties. Informally speaking, two spaces are identical from the perspective of topology if one space can be continuously deformed into the other without tearing or gluing. When a homeomorphism exists, the two spaces are considered *homeomorphic*.

Graphs or networks, are discrete topological spaces $\mathbb{G}$ with a finite set of elements V called *nodes* or vertices:

$$\mathbb{G} = (V, E).$$

The topological structure is defined via the set of *edges* E or connections that align the nodes with each other. The edges can also contain directional information in which case the graph is referred to as a directed graph or digraph.

Manifolds are n -dimensional, continuous spaces that are locally homeomorphic to the $\mathbb{R}^n$ but are globally topologically different from the $\mathbb{R}^n$. An n -manifold embedded in a higher-dimensional space like the $\mathbb{R}^m$ can be constructed via a characteristic function χ that identifies the points on the manifold:

$$\mathbb{M}^n = \{\mathbf{x} \in \mathbb{R}^m : 0 = \chi(\mathbf{x})\}.$$

Two classical examples are the n -sphere and the n -torus.

Sphere An n -sphere is a manifold defined as all points with a distance of 1 to the origin:

$$\mathbb{S}^n = \{\mathbf{x} \in \mathbb{R}^{n+1} : \|\mathbf{x}\| = 1\}.$$

Figure 1(a) illustrates the 1-sphere, the outline of a unit circle. Though its elements are two-dimensional points $[x, y]$, any continuous subset of the circle can be mapped onto a subset of the one-dimensional Euclidean space $\mathbb{R}^1$, the line. But there exists no mapping of the entire $\mathbb{S}^1$ without making a cut and therefore losing topological information. (b) shows the 2-sphere, the topological space of the Earth's surface. Famously, it requires an atlas—a collection of overlapping charts ($\mathbb{R}^2$)—for complete representation, as no single coordinate map can bijectively cover the entire surface.

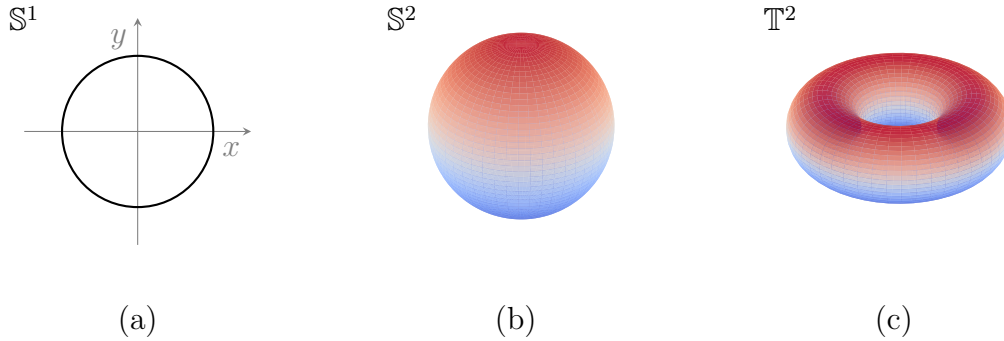


Figure 1: Examples of manifolds. (a) the 1-sphere, or a circle, (b) the 2-sphere, and (c) the 2-torus.

Torus The n -torus is yet another example of a manifold that is constructed as the Cartesian product of n circles:

$$\mathbb{T}^n = \underbrace{\mathbb{S}^1 \times \mathbb{S}^1 \dots \mathbb{S}^1}_{n \text{ times}}.$$

A two-dimensional point whose coordinates are both elements of distinct $\mathbb{S}^1$ is an element of the donut-shaped 2-torus visualized in Figure 1(c).

Angular Space An *angle* is a real number that measures rotation. When expressed in the unit radians [rad]—as done throughout this thesis—it is 2π -periodic, meaning any angle $a \in \mathbb{R}$ can be mapped to an angle a' that is an element of the angle space $\mathbb{A} = [-\pi, \pi)$ using the modulo function:

$$a' = \text{mod}(a + \pi, 2\pi) - \pi.$$

This periodization preserves all topological information while providing a unique representation of an angle. Continuous angular increase corresponds to traversal of the circle outline because the one-dimensional angular space $\mathbb{A}$ is homeomorphic to the $\mathbb{S}^1$. Higher dimensional angular spaces $\mathbb{A}^n$ are homeomorphic to the $\mathbb{T}^n$, where the two-dimensional case is sketched in Figure 2.

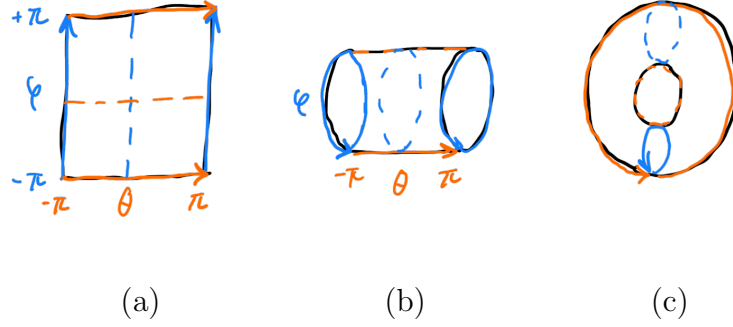


Figure 2: A two-dimensional angular space $[\theta, \varphi] \in \mathbb{A}^2$ is homeomorphic to the 2-torus. The sequence of sketches tries to visualize this relationship intuitively. (a) shows an incomplete mapping of the angular space $\mathbb{A}^2$ onto the $\mathbb{R}^2$. The colored axis arrows should be “glued” together to represent the periodicity of the angles. When the φ -axis is rolled to connect the θ -axes, the structure becomes a cylindrical tube as in (b). The torus appears when the cylinder is stretched and glued to connect the φ -axes as well (c).

3.2 Dynamical Systems

A *dynamical system* is a mathematical framework describing the evolution of a system over time. Formally, it is defined by the tuple

$$(T, S, \Phi),$$

where:

- T is the *time domain*, commonly $\mathbb{Z}$ or $\mathbb{R}$,
- S is the *state space* (or phase space),
- Φ is the *flow*[46].

The *state* vector $\mathbf{s} \in S$ encapsulates all time-varying properties of the system. A state at time $t \in T$ is denoted $\mathbf{s}(t)$, with $\mathbf{s}(0)$ as the *initial* state with $t = 0$. The state space S is the set of all possible states, often a subset of $\mathbb{R}^n$. The individual scalar coordinates of a state are referenced with indices: $\mathbf{s} = [s_0, s_1, \dots]$. The flow $\Phi : S \times T \rightarrow S$ describes how a state $\mathbf{s}(0)$ has evolved after time t :

$$\Phi(\mathbf{s}(0), t) = \mathbf{s}(t)$$

Dynamical systems are classified by their time domain, as shown in Figure 3:

1. *Continuous* systems as in (a) with $T = \mathbb{R}$ define Φ implicitly via a first-order differential equation:

$$f(\mathbf{s}(t)) = \dot{\mathbf{s}}(t),$$

where $\dot{\mathbf{s}}(t) = \frac{d}{dt}\mathbf{s}(t)$ is the time derivative of the state or the *velocity*. The function f is called the *vector field* of the system. If f does not depend on time, the system is *autonomous*.

2. *Discrete* systems as in (b) with $T = \mathbb{Z}$ define Φ implicitly via iterative map or difference equations which directly compute the state for the next time step:

$$F(\mathbf{s}(t)) = \mathbf{s}(t+1), \quad F: S \rightarrow S$$

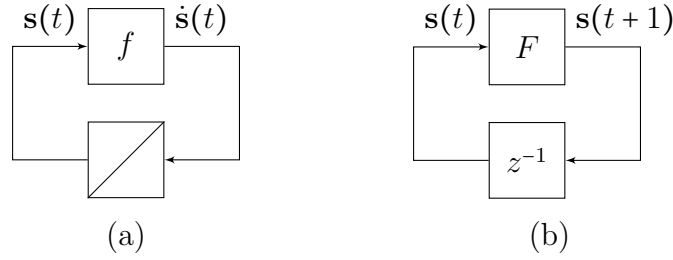


Figure 3: Flowcharts representing the two types of dynamical systems: (a) continuous systems with $t \in \mathbb{R}$, where $\square$ is an integrator, and (b) discrete systems with $t \in \mathbb{Z}$, with z^{-1} as a delay block.

Most physical systems are continuous but can be discretized for simulation, e. g., by using Euler's method:

$$F(\mathbf{s}(t)) \approx \mathbf{s}(t) + \Delta t f(\mathbf{s}(t))$$

The approximation gets more accurate the smaller Δt is chosen. Conversely, the velocity in discrete systems is approximated by:

$$\dot{\mathbf{s}}(t) \approx \frac{\Delta \mathbf{s}(t)}{\Delta t}, \quad \Delta \mathbf{s}(t) = \mathbf{s}(t) - \mathbf{s}(t-1).$$

A *trajectory* $\mathbf{s}(a:b)$ is the set of states that the system reaches within the time interval $[a, b]$:

$$\mathbf{s}(a:b) = \{\mathbf{s}(t)\}_{t \in [a, b]}$$

An *equilibrium* (fixed point or fixpoint) $\mathbf{s}_*$ is a state that satisfies:

$$\Phi(\mathbf{s}_*, t) = \mathbf{s}_*, \quad \forall t \in T.$$

In other words, an equilibrium is not affected by the dynamics of the system and does not change over time:

$$\mathbf{s}_* = F(\mathbf{s}_*) \quad \text{or} \quad \mathbf{0} = f(\mathbf{s}_*). \quad (1)$$

Equilibria are classified based on their stability properties.

3.2.1 Stability

Stability is most easily analyzed for autonomous linear systems, where $f(\mathbf{s})$ is expressed as a matrix multiplication with $\mathbf{J} \in \mathbb{R}^{n \times n}$:

$$\dot{\mathbf{s}} = \mathbf{J}\mathbf{s} \quad (2)$$

The eigendecomposition of $\mathbf{J}\mathbf{v}_i = \lambda_i\mathbf{v}_i$ yields the eigenvalues $\lambda_i \in \mathbb{C}$ and eigenvectors $\mathbf{v}_i \in \mathbb{C}^n$. The real part of the eigenvalues $\text{Re}(\lambda_i)$ determines the system's stability around the origin:

- $\text{Re}(\lambda_i) > 0$: Divergence along $\mathbf{v}_i$.
- $\text{Re}(\lambda_i) < 0$: Convergence along $\mathbf{v}_i$.
- $\text{Re}(\lambda_i) = 0$: No motion along $\mathbf{v}_i$.

The imaginary part $\text{Im}(\lambda_i)$ is responsible for rotation around the origin.

The four different kinds of equilibria for two-dimensional systems are visualized in Figure 4. For diagonal matrices, the eigenvectors are the matrix columns $\mathbf{v}_i = \mathbf{J}[i]$ and the eigenvalues are found on the main diagonal $\lambda_i = \mathbf{J}[i, i]$. The equilibria are qualitatively classified according to the eigenvalues at the origin of the system:

- $\mathbf{s}_\bullet$ *Stable fixpoint* : $\text{Re}(\lambda_i) \leq 0, \forall i$, all eigenvalues are negative (or zero). Trajectories converge (or do not move) in all directions as in (a). States of stable fixpoints are marked by a black dot $\bullet$.
- $\mathbf{s}_\circ$ *Unstable fixpoint*: $\text{Re}(\lambda_i) > 0, \forall i$, all eigenvalues are positive. Trajectories diverge in all directions, as in (d). States of unstable fixpoints are marked by a white-filled black circle $\circ$.

$\mathbf{s}_\bullet, \mathbf{s}_\circ$ *Saddle point*: $\text{Re}(\lambda_i)$ have mixed signs. Trajectories converge along the stable eigenvectors and diverge along the unstable ones, as in (b) and (c). The black half of the marker indicates the stable axis: For $\bullet$, s_0 is stable, and for $\circ$, s_1 is stable. For this thesis, this notation suffices as the considered systems are at most two-dimensional and the eigenvectors approximately align with the coordinate axes.

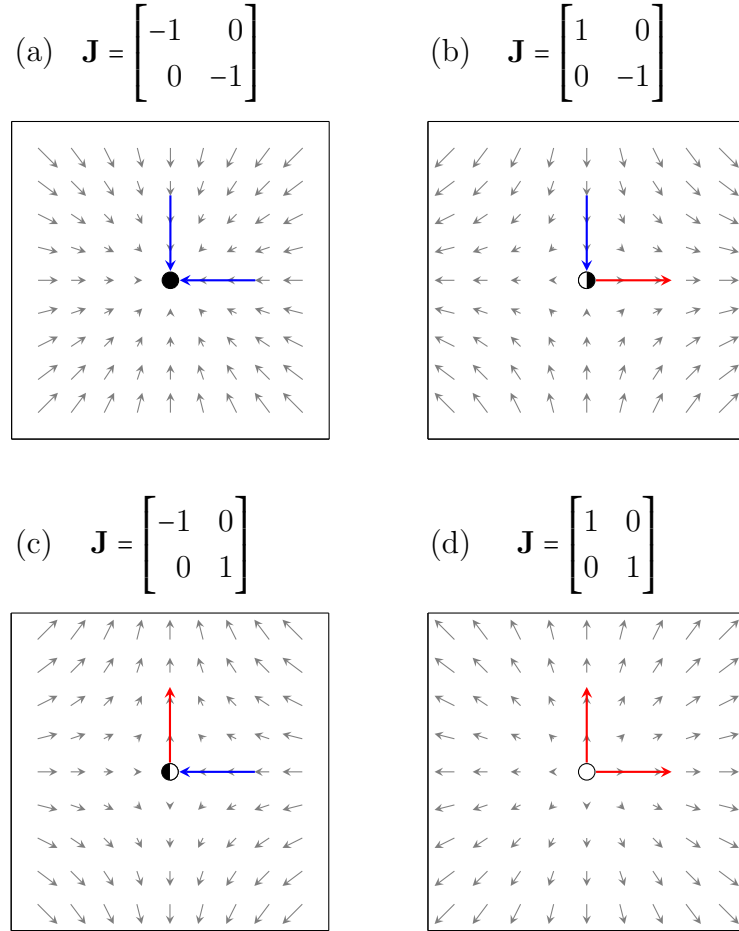


Figure 4: Vector fields of different autonomous linear dynamical systems with purely real eigenvalues. They feature the four fundamental types of equilibria: (a) Both eigenvalues are real and negative: a stable fixpoint, denoted as $\mathbf{s}_\bullet$. (b) and (c) One eigenvalue is positive, the other is negative: a saddle point, denoted as $\mathbf{s}_\bullet$ or $\mathbf{s}_\circ$. (d) Both eigenvalues are positive: an unstable fixpoint, denoted by $\mathbf{s}_\circ$. $\mathbf{v}_i$ are scaled and colored according to their respective eigenvalue λ_i .

According to the Hartman–Grobman theorem, the flow in a neighborhood close to the point of interest is qualitatively equivalent to its linearization. Which allows the application of linear stability analysis to a broader set of dynamic systems, namely those where f is at least once differentiable at the state of interest $\tilde{\mathbf{s}}$. These systems can be linearized at $\tilde{\mathbf{s}}$ using first-order multivariate Taylor-expansion:

$$\dot{\mathbf{s}} = f(\mathbf{s}) \approx f(\tilde{\mathbf{s}}) + \mathbf{J}_f(\tilde{\mathbf{s}})(\mathbf{s} - \tilde{\mathbf{s}})$$

where $\mathbf{J}_f(\tilde{\mathbf{s}})$ is the Jacobian matrix of f evaluated at $\tilde{\mathbf{s}}$ that holds all the partial derivatives:

$$\mathbf{J} = \left[\frac{\partial f(\tilde{\mathbf{s}})_i}{\partial s_j} \right]_{i,j \in [1,n]} = \left[\frac{\partial \dot{s}_i}{\partial s_j} \right]_{i,j \in [1,n]},$$

where s_i is the i -th component of the state vector $\tilde{\mathbf{s}}$. If the state of interest is chosen as an equilibrium $\tilde{\mathbf{s}} = \mathbf{s}_*$ then $f(\tilde{\mathbf{s}}) = \mathbf{0}$ holds (Eq. 1) and through shifting the origin of the system to $\tilde{\mathbf{s}}$ the linearization always yields an autonomous linear system just as in Eq. 2:

$$\dot{\mathbf{s}} = \mathbf{J}\mathbf{s}$$

This means that extracting the eigenvalues of the Jacobian matrix of the system function evaluated in a state of equilibrium $\mathbf{s}_*$ yields the stability characteristics of this equilibrium and allows its qualitative classification as a stable fixpoint, unstable fixpoint, or saddle point.

3.2.2 Attractors

An attractor $A \subset S$ is a subset of the state space that satisfies the following conditions:

1. A is invariant over time, meaning that

$$\text{if } \mathbf{s}(t) \in A, \quad \text{then } \mathbf{s}(t : \infty) \subset A$$

2. A has a *basin* $B(A) \subset S$ such that

$$B(A) \setminus A \neq \{\}.$$

A basin is the set of states that converge toward A for $t \rightarrow \infty$. This means that a state $\mathbf{s}(0) \in B(A) \setminus A$ will follow a trajectory where:

$$\mathbf{s}(0 : \tau) \subset B(A) \quad \text{and} \quad \mathbf{s}(\tau : \infty) \subset A.$$

States in the basin will stay within the basin. After a *time of convergence* τ , these states will be within the attractor.

3. A is *robust* to small disturbances. Meaning that there exists a δ -region:

$$U_\delta(A) = \bigcup_{\mathbf{s}_A \in A} \{\mathbf{s} \in S : \|\mathbf{s}_A - \mathbf{s}\| < \delta\}$$

around the attractor states, with an arbitrarily small $\delta > 0$ that is entirely contained by the attractor's basin $U_\delta(A) \subset B(A)$. Disturbed states would therefore converge back into the attractor.

4. A is minimal, such that no true subset of A satisfies the other conditions.

$A(\mathbf{s}(0))$ refers to the attractor to which the system converges considering the initial state $\mathbf{s}(0)$. A system's set of all attractors is denoted as:

$$\mathcal{A}(\Phi) = \{A_0, A_1, \dots\}.$$

Figure 5 visualizes the four different types of attractors:

1. *Stable fixpoint* $\mathbf{s}_\bullet$: A constant state satisfying

$$A(\mathbf{s}(0)) = \{\mathbf{s}_\bullet\}, \quad \mathbf{s}_\bullet = \lim_{t \rightarrow \infty} \Phi(\mathbf{s}(0), t).$$

2. *n-orbit*: A periodic trajectory with n states satisfying

$$A(\mathbf{s}(0)) = \{\mathbf{s}(\tau : \tau + n) \text{ and } \mathbf{s}(\tau + n) = \mathbf{s}(\tau)\}$$

A 1-orbit is actually a stable fixpoint. Orbits with different n are topologically different from each other.

3. *Quasi-periodic orbit*: Closed-loop oscillations that never exactly repeat but follow a recognizable pattern, as in (c).
4. *Chaotic attractor* or just chaos: unpredictable, non-repeating trajectories.

Saddle points and unstable fixpoints are—similar to attractors—invariant over time. Unlike attractors though, the basin of an unstable fixpoint $B(\{\mathbf{s}_\circ\}) = \{\mathbf{s}_\circ\}$ consists only of itself (condition 2). Because of the “repelling” effect that an unstable fixpoint has on its surrounding states it is also called a *repeller*.

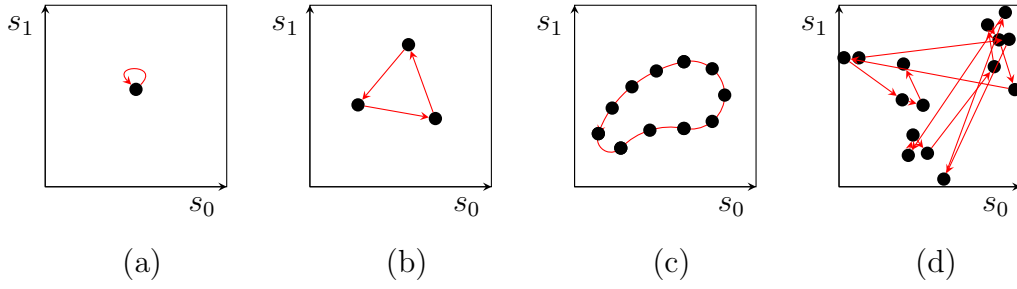


Figure 5: Dynamical systems can have four different kinds of attractors: (a) the stable fixpoint, an equilibrium state that trajectories converge toward; (b) an n -orbit, periodic trajectories of n states; (c) quasi-periodic orbits that oscillate but never exactly repeat; and (d) chaos.

Saddle points have a basin beyond themselves; states in the direction of the eigenvectors with negative eigenvalues, the so-called *stable manifold*. But saddle points—like unstable fixpoints—are not robust to noise (condition 3). Here, trajectories that experience small disturbances beyond the stable manifold are repelled.

A system with multiple attractors splits the state space into distinct basins. A boundary between two basins is called a *separatrix* and can be a smooth, continuous curve or have a fractal shape. Unstable fixpoints and saddle points both lie within these separatrices as shown in Figure 6.

3.2.3 Bifurcations

Dynamical systems may depend on a vector of parameters $\mathbf{u} \in U$. These values influence how the state changes but are themselves not changed by the system and hence not part of the state. To denote the dependency of the system dynamics on $\mathbf{u}$, it is written in the function index:

$$\Phi_{\mathbf{u}}(\mathbf{s}(0), t) \quad \text{or} \quad f_{\mathbf{u}}(\mathbf{s}(t)) \quad \text{or} \quad F_{\mathbf{u}}(\mathbf{s}(t)).$$

Changing a parameter may cause a state to converge to a different attractor

$$A_{\mathbf{u}}(\mathbf{s}_0) \neq A_{\mathbf{v}}(\mathbf{s}_0).$$

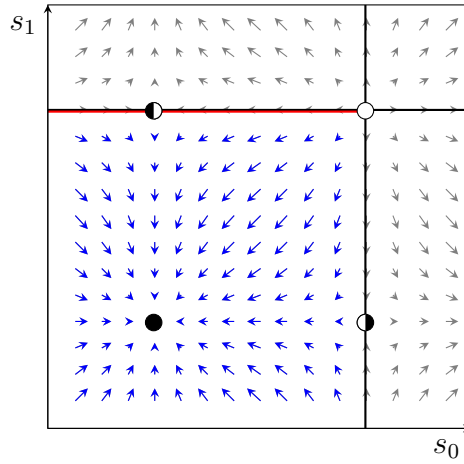


Figure 6: All states that converge toward the stable fixpoint $\mathbf{s}_\bullet$ make up the blue-colored basin $B(\mathbf{s}_\bullet)$. The border to other basins creates the separatrices (black lines) which house the unstable fixpoints $\mathbf{s}_\circ$ and the saddle points $\mathbf{s}_\bullet$ and $\mathbf{s}_\circ$. The red line shows the stable manifold of $\mathbf{s}_\bullet$. If not disturbed, trajectories on the separatrix travel along the stable manifold from the unstable fixpoint to the stable fixpoint.

There are three cases where $A_{\mathbf{u}} \neq A_{\mathbf{v}}$ holds:

1. The type of attractor changes. For example, a stable fixpoint starts to oscillate and becomes an n -orbit or a quasi-periodic attractor.
2. The original attractor disappears. For example, a stable fixpoint becomes unstable, and the state converges toward a different attractor. The new attractor might also be a stable fixpoint, but since the attractor landscape changed, it is topologically different.
3. A new attractor appears, toward which $\mathbf{s}(0)$ now converges instead. The new attractor might also be of the same type, but since the attractor landscape changed it is topologically different.

This topological change induced by a parameter variation is a *bifurcation*. The parameter values $\mathbf{u}$ that induce a bifurcation in $\mathbf{s} \in S$ are elements of the bifurcation set:

$$\mathcal{B}(\mathbf{s}) = \{\mathbf{u} \in U : \exists \Delta \mathbf{u} \text{ such that } A_{\mathbf{u}}(\mathbf{s}) \neq A_{\mathbf{u}+\Delta \mathbf{u}}(\mathbf{s})\},$$

where $\Delta \mathbf{u}$ is some small parameter change.

3.3 Sensorimotor Systems

A sensorimotor system is a type of dynamical system that consists of an agent that is embodied within a physical dynamic environment. The agent interacts with the environment via two channels:

1. *Motors*: The agent manipulates the state of the environment through a vector of motor signals $\mathbf{u} \in U$.
2. *Sensors*: The agent perceives the state of the environment through a vector of sensor signals $\mathbf{y} \in Y$.

3.3.1 Gravity

A concrete example of a dynamical system is a physical body that is embedded in a gravity-driven environment. Figure 7 introduces the robot of this thesis called *Brilliance*. It exists in a two-dimensional world where gravity accelerates downwards. The robot’s morphology consists of three limbs: two *leg* limbs connected to the left and right sides of a central *torso* limb. The joints are marked by colored circles in the figure: cyan $\circ$ for the left and orange $\circ$ for the right side. The joint angles θ_L and θ_R give the relative rotation between the respective leg and the torso. The tuple of all joint angles of a robot is also called *configuration* θ , where in Brilliance’s case:

$$\theta = [\theta_L, \theta_R] \in \mathbb{A}^2.$$

When the configuration angles are held constant at a fixed value and cannot be changed by the environment, the robot is essentially a rigid body.

The state of a rigid body in a two-dimensional world is given by its pose $\mathbf{s} = [x, y, \varphi]$. x and y are the coordinates of an arbitrary reference point of the body relative to the world origin, and φ is the orientation, i.e., the angle between the torso and the world. In the figure, the gray ground line serves as a reference for φ . Assuming that the terrain stays flat for any x and that gravity always adjusts y so that the body has ground contact, x and y can be omitted from the state:

$$\mathbf{s} = [\varphi] \in \mathbf{A}^1. \tag{3}$$

With this, changing the state effectively means rotating the robots’ entire body, as can be seen in Figure 7 from (c) to (d). The gravity-driven dynamical system will provide $\dot{\mathbf{s}} = \dot{\varphi}$, the angular velocity.

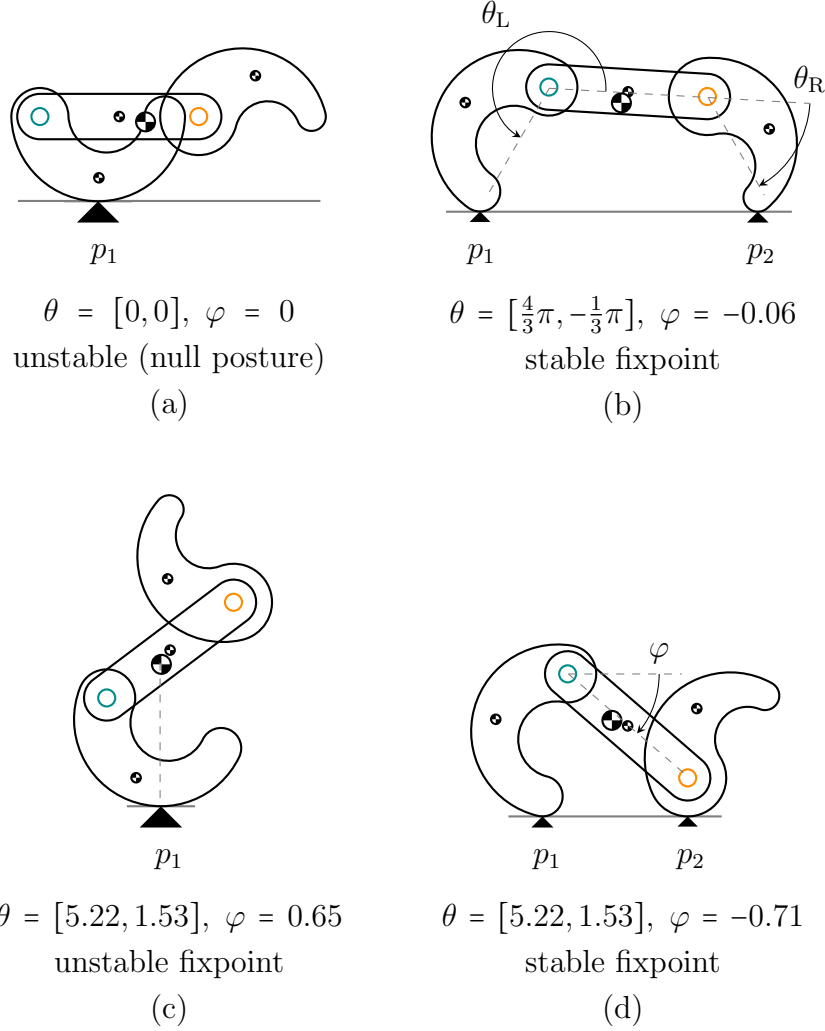


Figure 7: Selected poses of the Brilliance morphology. The morphology consists of three limbs that are connected by two joints marked by colored circles. The small $\bullet$ denotes the COM of the individual limbs whereas the big $\bullet$ is the combined COM. The scale of the GCPs marked by $\blacktriangle$ denotes the weight distribution. The different postures show the influence of the three angles θ_L , θ_R , and φ .

The dynamics of this system depend on two morphological properties:

- The point where Newtonian forces like gravity apply is called the *center of mass* (COM). Each limb has its own COM, marked by a small ●. The weighted sum of the individual COM gives the robot's combined COM $\mathbf{c} = [c_x, c_y]$, marked by a big ●.
- ▲ The coordinates where the body touches the ground are the *ground contact points* (GCP) $\mathbf{p}_i = [p_{xi}, p_{yi}]$ marked by a black, upwards pointing triangle ▲. Omitting y from the state (Eq. 3) guarantees that there is always at least one GCP $\mathbf{p}_1$. If more than one GCP exists they are indexed by i . Since the ground is at $y = 0$, $p_{yi} = 0$ for all GCP. The size of the GCP marker reflects the relative weight distribution.

Both $\mathbf{c}$ and $\mathbf{p}_i$ depend on and can be geometrically computed from the orientation φ (and later θ). The arrangement of their x -coordinates c_x and p_x determines the system dynamics $\dot{\varphi}$:

- If there is only one GCP and $c_x > p_x$ then $\dot{\varphi} < 0$. A negative rotation will take place as in Figure 7 (a).
- If there is only one GCP and $c_x < p_x$ then $\dot{\varphi} > 0$. A positive rotation will take place.
- If there is only one GCP and $c_x = p_x$ then $\dot{\varphi} = 0$. The state is an equilibrium. Either an unstable fixpoint like in (c) or a stable fixpoint with a curved contour touching the ground. If the curve's radius is larger than the distance $\|\mathbf{c} - \mathbf{p}_1\|$ it is stable.
- There is more than one GCP and

$$p_L \leq c_x \leq p_R \quad \text{then} \quad \dot{\varphi} = 0,$$

where p_L and p_R are the leftmost and rightmost GCP x -coordinates. In this case, the state is a stable fixpoint, as in (b) and (d).

The system vector field for continuous time could be modeled as

$$\dot{\varphi} = f_{\theta}(\varphi) = \omega \begin{cases} +1, & c_x < p_x \\ -1, & c_x > p_x \\ 0, & c_x = p_x \\ 0, & p_L \leq c_x \leq p_R \end{cases}, \quad (4)$$

with ω as some constant angular velocity.

Since $\varphi \in \mathbb{A}^1$, the state space is homeomorphic to the $\mathbb{S}^1$ and can be represented as such in Figure 8. With the configuration set to $\theta = [5.22, 1.53]$, three stable fixpoints $\varphi_\bullet$ emerge at -1.72 , -0.68 , and 2.62 , states where more than one GCP exists and c_x lies within the x -range of those. The unstable fixpoints $\varphi_\circ$ at -2.41 , -1.60 , and 0.69 are postures where the singular GCP vertically aligns with the COM. (The unstable fixpoint -1.60 lies too close to the stable fixpoint at -1.72 to be displayed in the figure.)

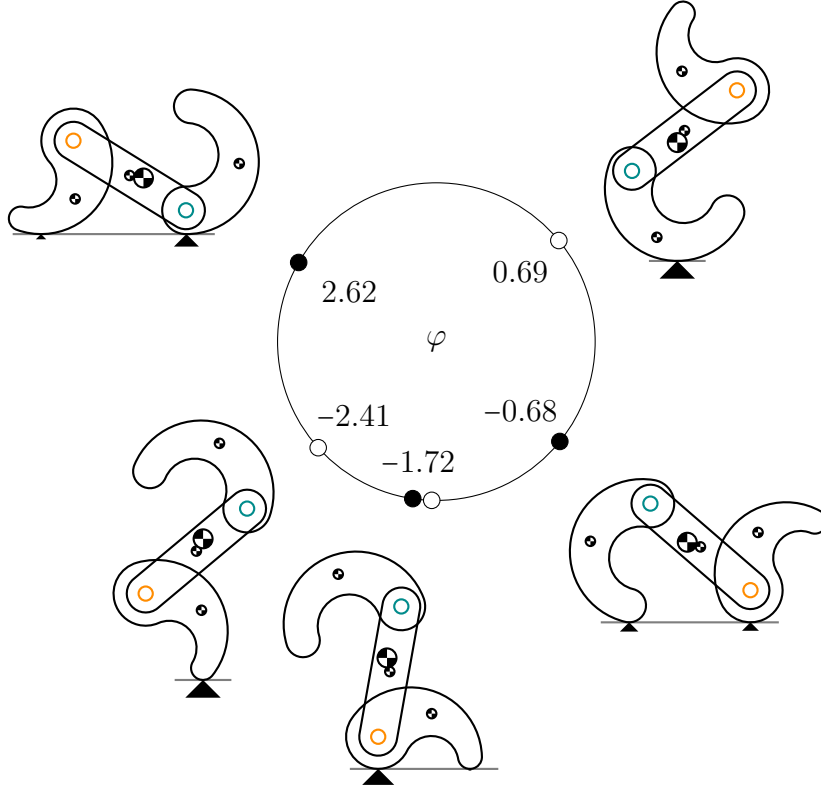


Figure 8: Circular visualization of the one-dimensional state space with $\mathbf{s} = \varphi$, the orientation angle between the torso limb and the world frame. Both joint angles are held constant at $\theta = [5.22, 1.53]$. Since φ is an angle, the state space is homeomorphic to a circle. The equilibrium poses that emerge due to gravity with this configuration are displayed with their respective stability marker: $\bullet$ for stable, $\circ$ for unstable.

3.3.2 Sensorimotor Manifolds

The sensorimotor manifold of a dynamical system is defined as:

$$\mathbb{M}(\Phi) = \{(\mathbf{y}, \mathbf{u}) \in Y \times U : \exists A \in \mathcal{A}(\Phi_{\mathbf{u}}), \mathbf{y} \in A, \mathbf{u} \notin \mathcal{B}(\mathbf{y})\}.$$

In other words, it contains all $(\mathbf{y}, \mathbf{u})$ -tuples where $\mathbf{y}$ is part of an attractor of the system given the parameters $\mathbf{u}$, but excluding any $\mathbf{u}$ that causes a bifurcation for the state $\mathbf{y}$. Bifurcations always lie on the border of stability and therefore do not locally resemble the Euclidean space. These manifolds topologically describe the sensorimotor relationship of the system.

In the example, treating the left leg angle as a mutable parameter to the system ($\mathbf{u} = [\theta_L]$)—while still keeping $\theta_R = 1.53$ constant—enables bifurcation analysis. Figure 9 depicts the bifurcation diagram of $f_{\theta_L}(\varphi)$. The y -axis shows the location of stable fixpoints $\varphi_{\bullet}$ depending on the parameter value θ_L on the x -axis. As both values are angles, the diagram is in both dimensions 2π -periodic and homeomorphic to the 2-torus ($\mathbb{T}^2$). While this toroidal structure is challenging to visualize directly in the planar representation, the angular periodicity implies that traversal beyond any diagram boundary corresponds to re-entry at the opposite boundary. The complete structure can be mentally reconstructed by imagining the diagram folded into a donut shape, following the homeomorphic mapping sketched out in Figure 2.

The state space from Figure 8 represents the gray vertical slice of this diagram at $\theta_L = -1.06$. The red arrows represent bifurcations. Here, a previously stable fixpoint becomes unstable due to varying θ_L . The arrows point at the stable fixpoint that the new state converges toward. States that align vertically have the same configuration and only differ according to their φ .

The stable fixpoints construct a one-dimensional manifold $\mathbb{M}(f_{\theta_L}(\varphi))$ embedded in this two-dimensional angular space. This manifold is unique to the morphology of the robot and can provide topological insights about its sensorimotor relationships:

1. The manifold can be classified into distinct *submanifolds*. All states within the same submanifold can be reached without falling. The postures that belong to the same submanifold are labeled with the same letter, e. g., (a1) and (a5) both belong to submanifold (a).
2. Bifurcations correspond to *falling* transitions. If the robot at state (a1)

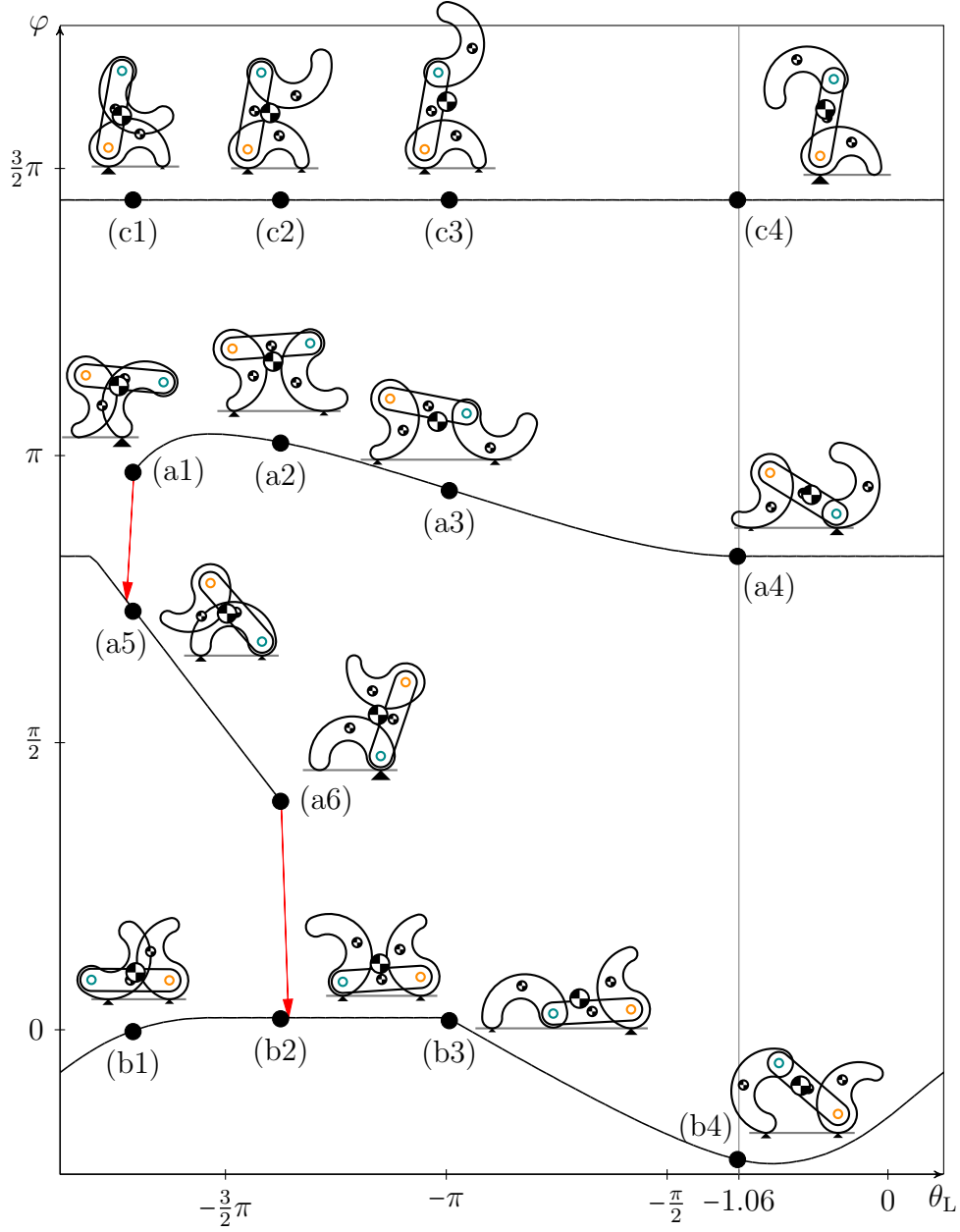


Figure 9: The bifurcation diagram of the Brilliance morphology sampled with $n = 120$ per axis. The black lines show how the state φ of the stable fixpoints changes in dependency on the parameter θ_L , the left leg angle. The right leg is held constant at $\theta_R = 1.53$. The red arrows visualize bifurcations: here, the parameter change induces a topological change, namely a stable state becomes unstable. This corresponds to a falling motion due to a small change in θ_L . The vertical gray line corresponds to the state space of Figure 8.

decides to decrease θ_L , it will eventually move its COM beyond the GCP- x -range and turn the stable fixpoint unstable. Now, φ will change or “fall” into a state similar to (a5).

3. Falling is *irreversible* or uni-directional. For example, the movement that transition (a1) to (a5) is a small negative rotation of the left joint and causes a large change in φ —a *fall*. However, the robot cannot apply the inverse, i.e., a small positive rotation, to transition from (a5) back to (a1).
4. The *accessibility* of different states can be inferred from the topology of the manifold. Even though there is no direct path from (a5) back to (a1), the robot can instead traverse the states (a4), (a3), and (a2) to reach (a1) again. While some submanifolds can be escaped—moving from (a6) to (b2)—others trap the robot, like (b) and (c) so that states on different submanifolds are inaccessible.
5. A *plateau* corresponds to no immediate impact from changing a parameter. Areas where the gradient $\partial\varphi/\partial\theta_L$ is zero create plateaus like around (a4), (b2), or the entire submanifold of (c). Here, the left leg does not touch the ground and hence has no influence on the position of any GCP.
6. A *crease* is the point where different smooth patches meet non-smoothly. This reflects a discontinuity in the gradient $\partial\varphi/\partial\theta_L$. Here, a limb is either being lifted off or placed on the ground. Normally, the GCP smoothly follows the outline of the limb while θ_L varies. When a new GCP gets introduced or a previous GCP is removed like in (b3) then there is a sudden change in the relationship of φ to θ_L , and a new smooth curve takes over.
7. *Curvature* corresponds to the number of different limbs that touch the ground. When only one limb supplies the GCP, as in (a5), (a6), or the entire submanifold of (c) then the lines are either constant or linear. When the lines have a curvature like around (a3) or (b4) two different limbs are on the ground.

Analogous to θ_L , the right leg angle θ_R is motorized and can therefore be treated as a parameter to the dynamical system. Now, $\mathbf{u} = \theta = [\theta_L, \theta_R]$. The

sensorimotor manifold $\mathbb{M}(f_\theta(\varphi))$ manifests as a collection of disconnected two-dimensional warped surfaces, embedded in the $\mathbb{A}^3$. Figure 10 shows Brilliance’s SMM embedded in an $\mathbb{R}^3$ projection. The three different azimuth angles allow for better spatial clarity. The colors correspond to the value of φ . The theoretically smooth manifold is here discretized and sampled at a resolution of 120 per parameter axis. Figure 9 represents the highlighted slice along the θ_L -axis.

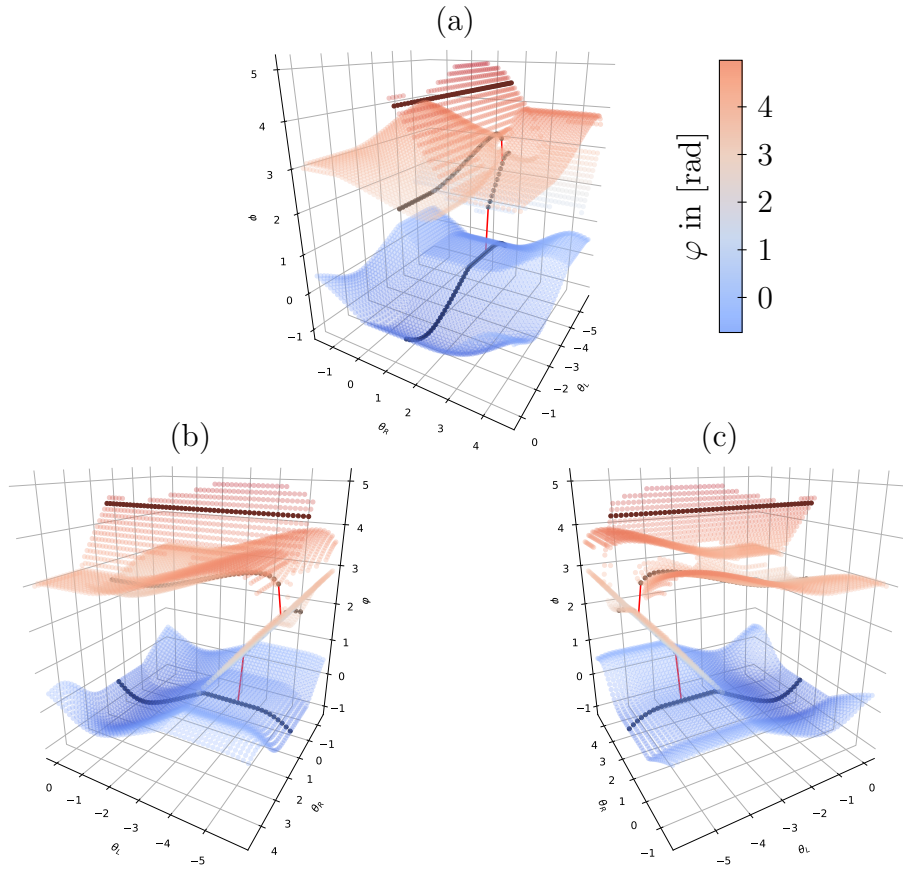


Figure 10: Brilliance’s three-dimensional sensorimotor manifold from three different perspectives. The number of samples is 50 per axis. The highlighted slice represents the bifurcation diagram from Figure 9. The coloring is according to φ on the z -axis.

3.3.3 Cognitive Sensorimotor Loop (CSL)

The cognitive sensorimotor loop (CSL) is an adaptive controller proposed by Hild[39] and designed to control the motor signal u of a rotational joint whose angle θ rotates with a rotational velocity $\dot{\theta}$. It directly couples the measured sensor value with the motor signal. The CSL control loop is depicted in the flowchart of Figure 11 and is defined as:

$$u(t+1) = g_f u(t) - g_i \dot{\theta}(t). \quad (5)$$

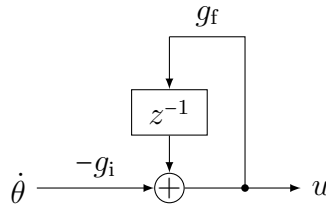


Figure 11: Flowchart of the Cognitive Sensorimotor Loop

It depends on two parameters: int input gain g_i and int feedback gain g_f . Different combinations of parameter values are called behavior *modes* and are categorized according to Table 1:

Table 1: CSL-modes and their parameters (adjusted from [44])

	Release	Hold	Contraction	Support
g_i	> 0	> 0	> 0	< 0
g_f	$\in [0, 1)$	$= 1$	> 1	$= 0$

- **Release-mode (r):** With $g_i > 0$, the input works against the movement of the joint. The motor passively follows outside forces like gravity and is essentially “relaxing” its joint. The delay unit and $0 < g_f \leq 1$ work as a leaky integrator that slows down movement proportionally to g_f . The release mode automatically finds stable fixpoints $\mathbf{s}_\bullet$. True relaxation can also be achieved by *disabling* the motors (coast mode) which is different from $u = 0$ when enabled (break mode). Breaking increases resistance to outside movement which results in larger dead-zones. In this thesis, r-mode is used synonymously with a disabled motor.

- **Hold-mode:** With $g_f = 1$, the exact integrator accumulates changes in angular value and will try to return to its initial position. This is equivalent to an I-controller.
- **Contraction-mode (c):** Further increasing $g_f > 1$ causes the CSL to actively work against outside forces. Deriving from eq. (5) with $\Delta u = 0$, the CSL balances output value and velocity such that

$$g_i \dot{\theta}(t) = (g_f - 1) u(t - 1).$$

When an outside force starts to slow down or speed up the motion the CSL adjusts u to work against this disturbance. The c-mode converges toward unstable fixpoints $\mathbf{s}_o$. Effectively, the stability characteristics of the system are inverted when in contraction. For c-mode to properly function, the parameters need to be tuned, e. g., as done in [43], otherwise it is prone to destabilize into uncontrolled oscillation. The c-mode in this thesis uses the parameters $g_i = 0.07$ and $g_f = 1.08$, with $\dot{\theta}$ measured in rad/s and $u \in [-1, 1]$ outputs a percentage of the maximum allowed voltage.

- **Support-mode:** A negative g_i creates a positive loop gain where u now operates in the same direction as $\dot{\theta}$. This causes the CSL to work with or “support” outside forces. Support-mode combined with a $g_f \in [0, 1]$ builds up an artificial momentum in the movement direction. This seems to be the intuitive solution to compensate for the dead-zone problem. Though in practice the support mode only reduces the probability of getting stuck in a dead-zone, it nonetheless fails to exit one on its own.

Using a CSL for controlling a rotational joint has a few advantages over conventional control:

- The CSL has no implementation overhead, is computationally cheap, and requires little memory. It can be implemented within a few lines of code, and even entirely analog implementations with only a few electronic components exist[41].
- No initial calibration is needed. Since the CSL works with the angular velocity, the absolute value of θ is irrelevant and does not have to be calibrated in advance.

- The CSL does not require explicit targets but intuitively finds its own target values by interacting with the environment. Depending on which mode is active the CSL transitions toward stable or unstable fixpoints.
- The target values found by a CSL correspond to energy-efficient postures. Stable fixpoints found in r-mode do not cost any energy to hold as the motor is simply turned off. Unstable fixpoints found in c-mode require some energy to reach, but once converged only little energy is needed to stabilize small disturbances.
- Since the CSL finds its own target values, robotic systems with multiple joints can control each joint independently. Communication is outsourced to the environment and the morphology. This allows for a decentralized control paradigm where each limb has its processing units like in the humanoid robot Myon[47].
- The CSL is designed for emergence. A simple sensorimotor coupling like the CSL can give rise to surprisingly complex behavior. As an example, a complex standing-up motion emerges within a robotic leg whose three joints are independently controlled by c-mode CSLs[40].

CSLs are themselves dynamical systems. Rather than being parameters, the configuration angles θ become system states along with φ when using CSL to control joints. g_i and g_f are the new system parameters. Defining a finite set of CSL modes means that the parameter space discretizes and the cardinality of possible actions to take significantly decreases. The dynamical system considered within this thesis therefore becomes:

$$\dot{\mathbf{s}} = f_m(\mathbf{s}), \quad \text{where} \quad \mathbf{s} = [\theta_L, \theta_R, \varphi] \in \mathbb{A}^3 \quad \text{and} \quad m \in \{r, c\}.$$

The exploration strategy called Attractor-based Behavior Control builds upon this concept.

3.4 Attractor-based Behavior Control

The Attractor-based Behavior Control[39] (ABC) is an exploration framework that systematically switches CSL modes to discover different equilibrium states of the dynamical system by only traveling along *heteroclinic orbits*. The otherwise continuous state space is therefore effectively discretized to only a few

equilibria per degree of freedom (DOF). This enables a sparse but nonetheless topologically rich representation of the sensorimotor manifold.

3.4.1 A Toy Example

A simple two-dimensional dynamical system serves as a playground to demonstrate the principles of the ABC. It is defined on the state space $S = \mathbb{R}^2$, with the state vector as $\mathbf{s} = [\theta_L, \theta_R]$ and the following dynamics:

$$\dot{\theta}_L = -\cos(\theta_L) \quad \text{and} \quad \dot{\theta}_R = -\cos(\theta_R).$$

This system can be visualized as a potential landscape, where the height $z(\theta_L, \theta_R)$ corresponds to the integral of the dynamics:

$$z(\theta_L, \theta_R) = \sin(\theta_L) + \sin(\theta_R).$$

Figure 12 (a) shows the landscape as a surface in $\mathbb{R}^3$, while (b) displays both the equipotential contours and the associated vector field representing the system's dynamics $f(\mathbf{s}) = -\nabla z$ as the gradient.

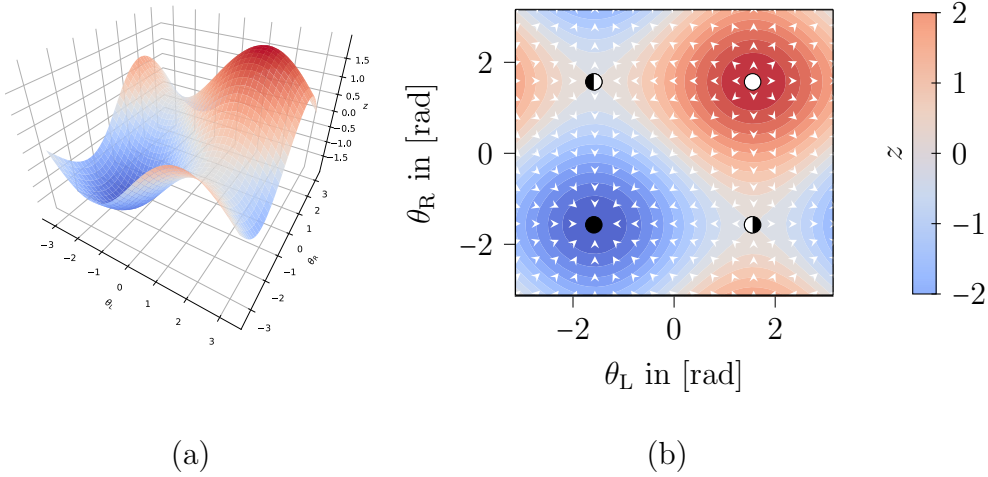


Figure 12: A toy dynamical system in the form of a surface $z(\theta_L, \theta_R) = \sin(\theta_L) + \sin(\theta_R)$ whose negative gradient $-\nabla z = f(\mathbf{s})$ is the system's vector field. (a) shows the three-dimensional surface of z , whereas (b) depicts the state space with z visualized as contour lines and f as a vector field.

This landscape provides an intuitive visualization of the system's behavior. A mass placed on this landscape would naturally roll downhill toward the

nearest local minimum (valleys) and locate a stable fixpoint, mimicking the system's trajectory under gradient descent or in r-mode control. Conversely, an air bubble trapped under the surface in water would float toward the nearest local maximum (peak) and locate an unstable fixpoint, which corresponds to the c-mode controlled trajectory. This can also be achieved by multiplying the corresponding velocity component by -1 and turning gradient descent into gradient ascent. Independent r-mode and c-mode control along different axes enables the targeting of saddle points, where the system exhibits stable dynamics along one eigenvector and unstable along another.

The following notation represents the current *state mode* or CSL modes of a two-dimensional system:

$$\mathbf{m} = (m_L, m_R) \quad \text{with} \quad m_L, m_R \in \{r, c\},$$

where m_L and m_R are the left and right joint's CSL modes. Table 2 lists all combinations of state modes and which equilibrium the corresponding trajectories converge toward.

Table 2: State modes and related equilibria for a two-dimensional systems

Symbol	$\mathbf{m}$	Equilibrium
●	(r,r)	stable fixpoint
○	(c,c)	unstable fixpoint
◐	(r,c)	saddle point (left is stable)
◑	(c,r)	saddle point (right is stable)

3.4.2 Heteroclinic Orbits

A *heteroclinic orbit* is defined as the trajectory that joins two different equilibria $\mathbf{s}_{*i}$ and $\mathbf{s}_{*j}$.

$$(\mathbf{s}_{*i} \rightarrow \mathbf{s}_{*j}) = \mathbf{s}(t_i : t_j) \quad \text{where} \quad \mathbf{s}(t_i) = \mathbf{s}_{*i} \quad \text{and} \quad \mathbf{s}(t_j) = \mathbf{s}_{*j}.$$

An instantaneous change of the CSL-mode from contraction to release or vice versa is also called *switching*. Switching while residing in an equilibrium evidently transitions along a heteroclinic orbit. The new CSL mode repels the

current state and instead attracts a different type of equilibrium:

$$\begin{array}{lll} r \rightarrow c & \text{corresponds to} & s_{\bullet} \rightarrow s_{\circ} \\ c \rightarrow r & \text{corresponds to} & s_{\circ} \rightarrow s_{\bullet} \end{array}$$

The ABC exploration only switches the mode of a single joint. This approach offers two key advantages:

1. Non-switched, passive joints remain near their equilibrium states throughout the transition. Release-mode-controlled joints passively maintain stability, while c-mode-controlled joints require only minimal energy input to compensate for instability.
2. With a single active variable the system is less prone to disturbances. Mode switches applied to specific states are repeatable allowing for reliable traversal of the SMM—given the symmetry is properly broken.

3.4.3 Breaking the Symmetry

When switching CSL modes within an equilibrium state, the system exhibits a fundamental symmetry: the transition may occur with equal probability in either direction along the state axis, positively (+) or negatively (−). Without further control, the resulting direction depends on random noise, so this symmetry must be deliberately broken to achieve predictable and reliable transitions.

Different methods to accomplish this exist. [39] introduces directional preference through a constant bias in the motor signal. This augments the mode space to $\{r+, r-, c+, c-\}$. Another option is the kick-fly-catch paradigm[43] where an initial high-gain impulse induces motion in the desired direction (kick), then the passive dynamics carry the movement by disabling the motor and saving energy (fly), and finally the c-mode controlled CSL stabilizes the state at the target equilibrium (catch).

Building on these methods, I formalize a directional switching notation. The switch mode $\mathbf{d}$ of a two-dimensional system is defined as:

$$\mathbf{d} = (d_L, d_R) \quad \text{with} \quad d_L, d_R \in \{+, -, \cdot\},$$

where d_L and d_R are the desired movement directions of the left and right joints respectively. They can be:

+, meaning *active*, positive movement,

−, meaning *active*, negative movement, or

·, meaning no active change for this joint, it stays *passive*

To avoid confusion within this thesis, the term “passive” refers to non-switching joints, and the term “relaxed” refers to r-mode controlled joints. A joint might be actively switched into a relaxed r-mode state and is therefore *not* passive.

The following notation represents a heteroclinic orbit from state mode $\mathbf{m}_i$ to $\mathbf{m}_j$ that is generated by a switch $\mathbf{d}$:

$$\mathbf{m}_i \xrightarrow{\mathbf{d}} \mathbf{m}_j \quad \text{or} \quad \mathbf{s}_{*i} \xrightarrow{\mathbf{d}} \mathbf{s}_{*j}.$$

The active directional component dictates which CSL mode changes.

3.4.4 ABC Graph Structure

Within the ABC framework, only one joint undergoes active switching, while others remain passive (non-switching). For a system with n degrees of freedom (DOF), the number of possible transitions grows linearly as $2n$. The two-dimensional case yields four possible switch modes:

$$\mathbf{d} \in D = \{(+, \cdot), (-, \cdot), (\cdot, +), (\cdot, -)\}$$

Given an ABC-based exploration, a directed graph structure $\mathbb{G}$ emerges across the state space:

$$\begin{aligned} \mathbb{G} &= (V, E) \\ V &= \{v_i = (\mathbf{s}_{i*}, \mathbf{m}_i) \in S \times \{\mathbf{r}, \mathbf{c}\}\} \\ E &= \{e_{ij} = (v_i, v_j, \mathbf{d}) \in V \times V \times D : v_i \xrightarrow{\mathbf{d}} v_j\} \end{aligned}$$

A node $v_i \in V$ represents an equilibrium $\mathbf{s}_{i*}$ and its corresponding state mode $\mathbf{m}_i$. The node also refers to a specific body posture denoted with capital letters (A, B, C, etc.). The directed edges $e_{ij} \in E$ encode the heteroclinic orbits $v_i \xrightarrow{\mathbf{d}} v_j$ between two nodes induced by applying a $\mathbf{d}$ -switch to v_i . Each node has $2n$ (here 4) of such outgoing edges according to the number of possible switch modes. In a theoretical unbounded, non-bifurcating system, this creates the periodic lattice pattern with alternating stability properties of Figure 13.

space.

Within the ABC graph, the stall can be represented as self-referencing loops:

$$\mathbf{m}_i \xrightarrow{\mathbf{d}} \mathbf{m}_i \quad \text{or} \quad \mathbf{m}_i \circlearrowleft^{\mathbf{d}}.$$

4 Robotic Platform

This section describes the robotic platform designed to implement and develop the concepts presented in this thesis. The physical system is called *Brilliance* and is depicted in Figure 14(a), with a corresponding schematic representation of its planar geometry provided in (b).

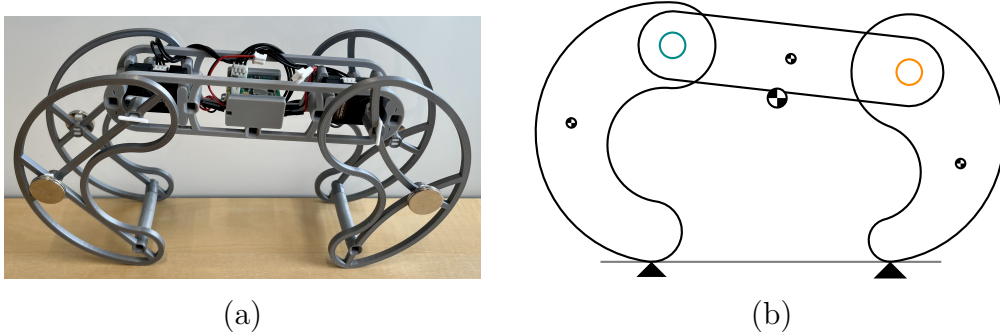


Figure 14: (a) a photo of the physical robot used in the thesis called Brilliance. (b) a schematic representation of the morphology that contains relevant geometrical markers such as COM ($\oplus$), GCP ($\blacktriangle$), and joints ($\circ$, $\circ$).

The following subsections systematically detail the experimental apparatus and methodology. First, I present the complete hardware specifications and electronic components at the heart of the robot. Next, I examine the mechanical design, manufacturing process, and physical assembly of the morphology. Finally, I outline the offline computational pipeline developed for construction of the SMM and visualization of collected sensorimotor data through animation and graphical representations featured throughout this work.

4.1 Hardware and Electronics

Motors The robotic platform employs two *Dynamixel XL330-M288-T* motors at each joint. Figure 15 (a) displays a photo of these lightweight, compact actuators. The motors operate in voltage control mode where the PWM duty cycle can be directly set. They provide real-time feedback of joint angle θ and angular velocity $\dot{\theta}$. This way, the CSL can directly control the motor voltage output $u(t)$. The current $i(t)$ is measured at a resolution of 1 mA and enables monitoring of the robot's energy consumption. Table 3 lists the essential specifications. Further details can be found in the motor's manual[48].

Table 3: Specifications of Dynamixel XL330-M288-T motors

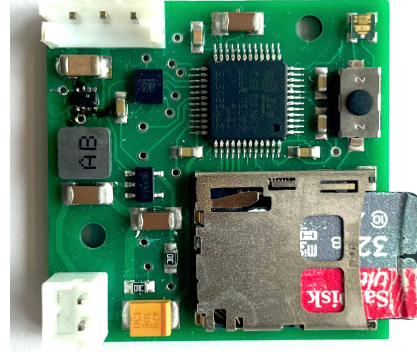
Parameter	Value
Dimensions	20 mm $\times$ 34 mm $\times$ 26 mm
Weight	18 g
Gear ratio	288.4 : 1
Recommended voltage	5 V
Stall torque	0.52 N m
No-load speed	103 rev/min
Current resolution	1 mA
Position resolution	0.088°
Velocity resolution	0.229 rev/min
PWM control resolution	0.113% resolution

Processor The *AccelBoard6D* (AB6D) is a newly developed processing unit designed to enhance sensorimotor computation. The circuit board seen in Figure 15 (b) has compact dimensions of 30 mm $\times$ 30 mm. At its core, the system integrates an STM32U575 microcontroller from STMicroelectronics, featuring a 32-bit ARM Cortex-M33. This processor provides 2 MB of flash memory, 786 kB of SRAM, and a range of hardware accelerators and peripheral support. The AB6D features an inertial measurement unit (IMU), combining a 3-axis accelerometer and a 3-axis gyroscope. The system incorporates a μ SD card slot which enables non-volatile storage of sensorimotor data during operation and boot loading of behavior programs. For this thesis, the software is developed in C using the STM32CubeIDE[49]. For comprehensive details on the board’s design and technical details, refer to [50]. The main loop is timed at a frequency of $f_s = 100$ Hz, including sensor sampling, algorithmic updates, and motor commands.

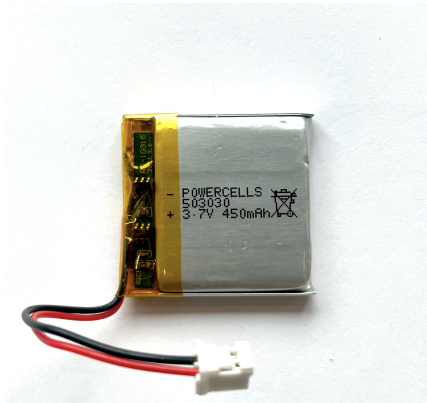
Battery The system is powered by the compact lithium-polymer battery in (c) with a nominal voltage of 3.7 V. Its physical dimensions of 30 mm $\times$ 30 mm align precisely with the AB6D circuit board. Voltage regulation is achieved via the MT3608 DC-DC step-up converter module from (d), which elevates the battery output to the required 5 V operating voltage for both the motor and processing unit.



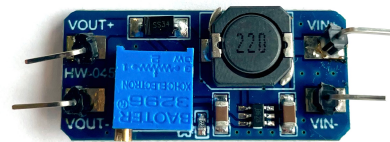
(a)



(b)



(c)



(d)

Figure 15: The hardware and electronic components used for the robot are (a) the motors, (b) the processor circuit board, (c) the battery, and (d) the step-up voltage converter.

4.2 Morphology

Morphology refers to the robot’s physical structure and shape. The design approach simulates a planar robotic system through careful component arrangement: all motors are aligned to restrict rotational motion to the sagittal xy -plane, while the limb geometries are linearly extruded along the z -axis to create functional three-dimensional structures. Figure 16 depicts the design process for the right leg limb.

The limb geometry is designed with the *arc representation*, a parametric modeling method that extends conventional polygon-based approaches. It constructs smooth, continuous limb outlines by tangentially concatenating circular arcs. Simple linear connections are also possible when the curvature parameter is set to zero. Figure 16 (a) illustrates the parametrized arc shape called *Cashew*, defined by four key parameters:

d : Distance between left and right arc centers

r_1, r_2 : Radii of the left and right arcs respectively

γ : Central angle of the connecting arcs which determines their curvature (with $\gamma = 0$ yielding linear connections)

This parametric model is transformed into STL meshes containing interface elements to connect to other parts (b), enabling rapid fabrication using standard 3D printing technology with a print time of approximately 2 h per limb. Final assembly utilizes a tool-free plug-and-clamp system (c). The complete mathematical framework and design methodology for this arc-based representation are detailed in prior work[51].

In order for the c-mode to properly react to gravitational pull the limbs need to have a significant mass. Since the PLA filament is relatively lightweight, magnets allow the customization of the individual limb mass. With four magnets of about 20 g per piece attached to the side, one leg limb has a mass of 45 g. The torso limb including all electronic components, weighs 100 g. The fully assembled robot therefore has a total mass of 190 g.

Table 4 presents the complete parameter set for the Brilliance morphology. Two important factors influenced the decision to use this particular body shape. First, comparative locomotion studies[33] demonstrated that this morphology achieves faster locomotion among four candidate designs. Second,

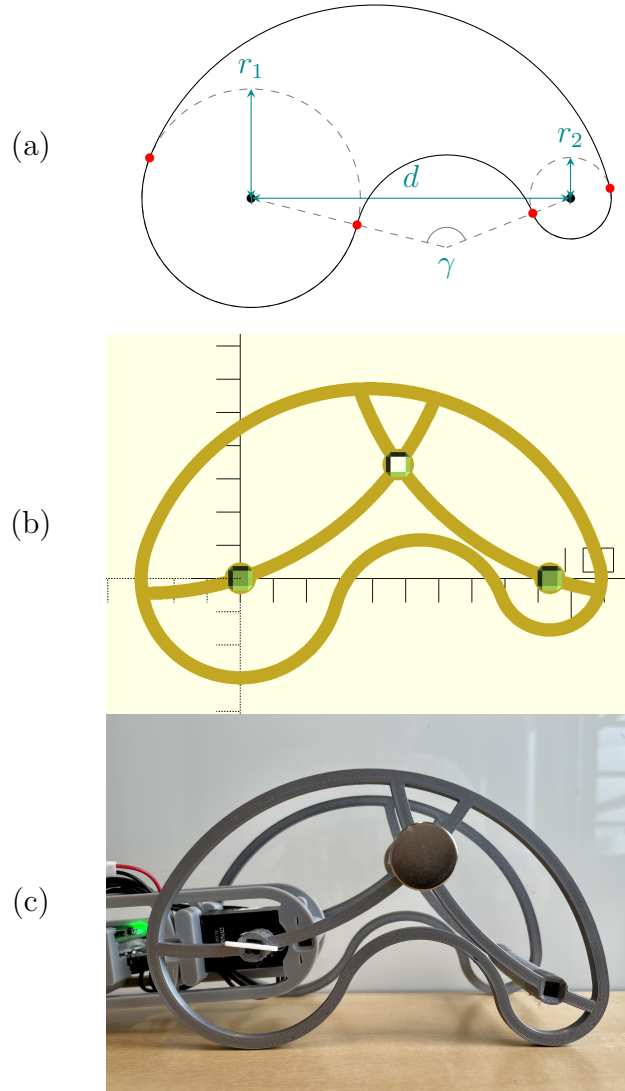


Figure 16: The design pipeline of Brilliance’s right leg limb. (a) shows the mathematical schematic of the underlying arc shape. This shape is then modeled into a printable 3D mesh (b) including connectors for the plug-and-clamp kit. The physical limb on the fully assembled robot can be seen in (c).

Table 4: Parameters of the Brilliance morphology (1u = 18.8132 mm)

	Limb	d [u]	r_1 [u]	r_2 [u]	γ [rad]
	Torso	7	1	1	0
○	Left leg	5.506 09	1.258 59	0.890 901	3.071 62
○	Right leg	4.969 73	1.6988	0.633 274	−2.516 11

when positioned as shown in Figure 14, the morphology is constrained to its current submanifold.

Said manifold is displayed in Figure 17. (a) shows the three-dimensional surface in the $[\theta_L, \theta_R, \varphi]$ -space, whereas (b) offers a two-dimensional projection onto the configuration space with φ as colored contour lines. This constraint to the submanifold offers several practical advantages:

1. Non-overlapping submanifolds can unambiguously be projected from (a) to the two-dimensional configuration space (b) improving visualization.
2. The morphology is prevented from toppling over. A falling impact introduces large physical forces on the body which could potentially damage the robotic platform.
3. Any heteroclinic transition can be assumed as bidirectional. As the inverse transition does not have to be tested, the exploration process is simplified.
4. Normally, φ is needed to disambiguate postures of similar configuration. But when constrained to a non-overlapping submanifold, φ can be ne-

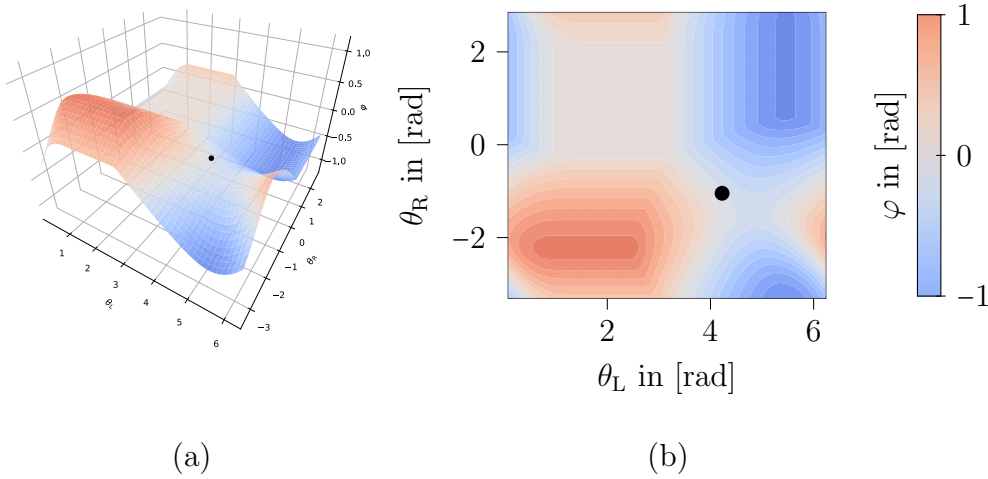


Figure 17: Within this thesis, the Brilliance morphology is constrained to this submanifold. (a) shows the submanifold as a surface in the three-dimensional state space. This surface can be projected onto the θ -configuration space as in (b). The black marker shows the state that reflects the robot's posture from Figure 14.

glected further reducing system complexity. The IMU data does not have to be processed.

4.3 Simulation Framework

I developed a custom software framework to support essential offline computation and analysis tasks for this thesis. The core functionalities are:

1. Mathematical modeling and visualization of robot morphologies as described above.
2. It implements a stabilization algorithm that computes the equilibrium body orientation φ' for any given state $[\theta_L, \theta_R, \varphi]$ using the rigid body dynamics model specified in Equation 4.
3. The computation of the SMM is achieved through exhaustive state space iteration. It constructs a directed graph representation of all stable states at a given resolution, serving as a ground truth reference for physical system validation or for visualization such as Figure 17.
4. The framework processes and visualizes sensorimotor data recorded during robot operation. This includes the reconstruction of the exploration graph generated by the ABC algorithm's online execution and possible merging of exploration graphs from multiple experimental runs at different initial poses.
5. Finally, it supports locomotion planning by analyzing the explore graphs and synthesizing locomotion cycles.

5 Self-Exploration

This section presents the practical implementation of the Attractor-Based Control (ABC) for the Brilliance morphology. The exploration algorithm illustrated in Figure 18 as a flowchart initiates from a fully-relaxed (r,r) posture $\mathbf{s}_{\bullet 0}$. This state forms the first node of the exploration graph $\mathbb{G}$.

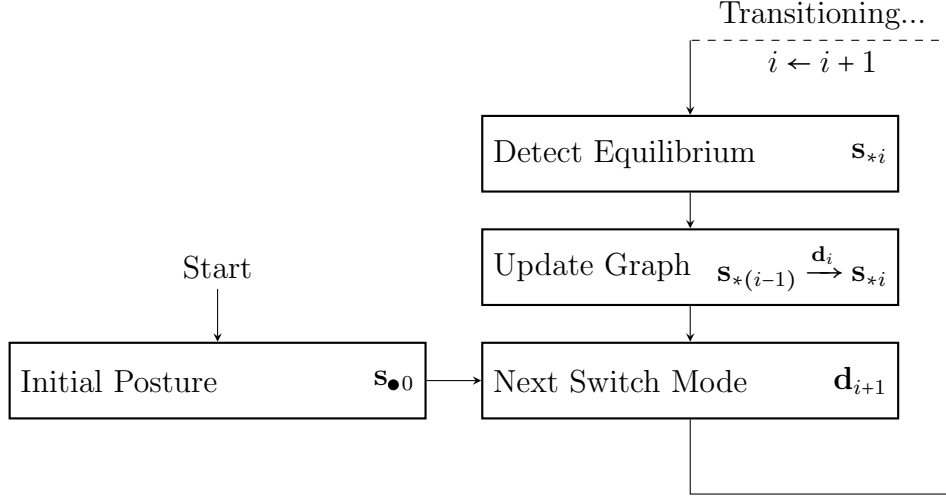


Figure 18: The ABC-based exploration’s update function as a flowchart.

The exploration process follows an iterative cycle of these steps:

1. The new switch mode $\mathbf{d}_i$ changes the CSL mode of one of the joints.
2. Transition along heteroclinic orbit $\mathbf{s}_{*(i-1)} \xrightarrow{\mathbf{d}_i} \mathbf{s}_{*i}$ occurs.
3. The algorithm waits until an equilibrium state is detected.
4. The new node and edge are added to $\mathbb{G}$.

The subsequent subsections detail the individual steps of this process. Firstly, I present an equilibrium detection system capable of identifying both fixpoint stability (r-mode) and oscillations (c-mode) from a single input sensor stream. Next, I analyze the *dead-zone* phenomena, discussing both its challenges for an ABC implementation and its potential utility through asymmetric exploitation. Finally, I evaluate the algorithm’s performance on the physical robot with experiments, paying particular attention to visualization of the emergent topological patterns.

5.1 Equilibrium Detection

The equilibrium detection algorithm processes a continuous stream of joint angle measurements $\theta(t)$ to identify when the system reaches either (1) a stable fixpoint $\theta_\bullet$ characteristic of r-mode convergence or (2) an unstable fixpoint $\theta_\circ$ around which c-mode controlled joints typically oscillate.

Before equilibrium detection, the raw joint angle measurements $\theta(t)$ undergo exponential smoothing to filter high-frequency noise while preserving the underlying low-frequency dynamical movements. The discrete IIR-filter implements:

$$\tilde{\theta}(t) = \begin{cases} \theta(0), & t = 0 \\ (1 - \alpha) \tilde{\theta}(t-1) + \alpha\theta(t), & t > 0 \end{cases}$$

where $\alpha \in [0, 1]$ represents the smoothing factor. The unfiltered adoption of the initial value $\tilde{\theta}(0) = \theta(0)$ prevents the introduction of startup biases.

5.1.1 Fixpoint Detection

For joints under r-mode control, the Fixpoint Detection (FPD) algorithm identifies asymptotic convergence by monitoring angular velocity $\dot{\theta}(t)$. Within a stable fixpoint $\dot{\theta} = f(\theta_\bullet) = 0$ holds, so the detection criterion requires the system to maintain its absolute difference $\Delta\theta$ of consecutive angle measurements within a small tolerance threshold $\Delta\theta_T$ for a minimum duration $t_{\max}$.

The FPD parameters and variables as well as the loop-wise update function are given in Algorithm 1. Each new filtered measurement $\tilde{\theta}(t)$ provides the current angular difference:

$$\Delta\theta = \tilde{\theta} - \theta_{\text{old}} \approx \Delta t \dot{\theta}.$$

A convergence counter t_T increments when $|\Delta\theta| < \Delta\theta_T$ and resets to zero otherwise. The detection condition is mathematically expressed as:

$$t_T \geq t_{\max}.$$

Upon convergence, the algorithm returns $\theta_\bullet = \theta_{\text{old}}$ as the estimated fixpoint. Smaller $\Delta\theta_T$ reduces steady-state error but increases detection time. Smaller $t_{\max}$ accelerates detection but raises false-positive risk.

Figure 19 demonstrates the FPD algorithm's performance during an exemplary state transition recorded on the physical robotic platform. The system

Algorithm 1 Fixpoint Detector (FPD)

<i>Parameters</i>		Default
$\Delta\theta_T$	Angular difference tolerance	1×10^{-4} rad
$t_{\max}$	Number of time steps to detect a fixpoint	50
<i>Variables</i>		Initial Value
t	Time step counter	0
θ_{old}	Sensor value from last call	0
t_T	Time steps with $ v < \Delta\theta_T$	0
UPDATE($\tilde{\theta}$)		
1: if $t > 0$ then		
2: $\Delta\theta = \tilde{\theta} - \theta_{\text{old}}$		
3: if $ \Delta\theta < \Delta\theta_T$ then		
4: $t_T \leftarrow t_T + 1$		
5: else		
6: $t_T \leftarrow 0$		
7: end if		
8: end if		
9: $\theta_{\text{old}} \leftarrow \tilde{\theta}$		
10: $t \leftarrow t + 1$		
DETECT()		
1: return $(t_T \geq t_{\max})$		
EQUILIBRIUM()		
1: return θ_{old}		

begins at $t = 0$ in a fully relaxed (r,r) configuration. After the $(+, \cdot)$ -switch command, the following dynamics are observed: the left leg enters c-mode control rotating against the ground and pulling the left joint up. The right leg passively remains in r-mode, and θ_R undergoes positive rotation due to ground interaction. The velocity profile $\dot{\theta}_R(t)$ shows asymptotic convergence with the FPD algorithm triggering its first detection at $t = 12.12$ s estimated fixpoint of $\theta_\bullet = 0.436$ rad. As the left leg keeps actively stabilizing its position the resulting oscillation indirectly transfers to the right leg, which is why the detection criterion occasionally resets.

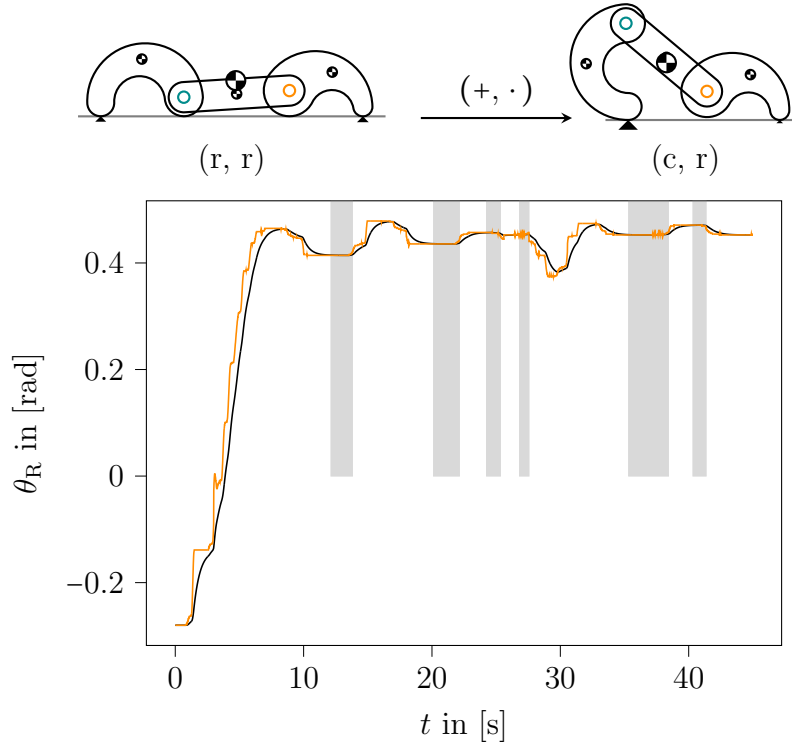


Figure 19: During the heteroclinic orbit transition $(r, r) \xrightarrow{(+, \cdot)} (c, r)$ seen here, the right leg angle θ_R is passively rotated due to ground contact and converges toward the stable state $\theta_\bullet \approx 0.436$ rad, with raw θ_R in orange, smoothed $\tilde{\theta}_R$ in black, and the gray regions indicating where a detection ($t_T \leq t_{\max}$) happens. The FPD confirms the first detection at $t = 12.12$ s. Even within the stable fixpoint, the right leg angle experiences some disturbances due to indirect motion transfer from the active left leg. (sampling frequency $f_s = 100$ Hz, smoothing factor $\alpha = 0.02$)

5.1.2 Oscillation Detection

In c-mode, the system exhibits qualitatively different behavior. While theoretically the c-mode CSL should maintain perfect equilibrium at the unstable fixpoint θ_o with $\dot{\theta} = 0$, practical implementation reveals more complex behavior. Hardware latency in the control loop, combined with inevitable environmental perturbations, induces persistent oscillations about the true equilibrium. These oscillations manifest as irregular, quasi-periodic signals characterized by alternating local maxima (peaks) and minima (valleys), corresponding to directional changes in the angular difference across time. The Oscillation Detection (OCD) algorithm identifies these velocity reversals, with its formal parametrization and loop-wise update function specified in Algorithm 2.

The angular difference $\Delta\theta$ is computed identically to the FPD case, using the difference of subsequent smoothed angle measurements. Maxima occur when $\Delta\theta$ transitions from positive to negative whereas minima correspond to negative-to-positive transitions. The OCD algorithm maintains a variable $a \in \{-1, 0, 1\}$ representing the previous velocity direction¹. A velocity reversal is identified when:

$$\Delta\theta a < 0.$$

Opposing signs indicate a change in direction between consecutive velocities.

The unstable equilibrium position θ_o is estimated through averaging of oscillation extrema. The estimation process builds upon the cumulative averaging rule:

$$\theta_{\text{avg}} \leftarrow \frac{(n-1)\theta_{\text{avg}} + \tilde{\theta}}{n} \quad \text{and} \quad n \leftarrow n+1,$$

where the average estimate θ_{avg} is updated with every new sample $\tilde{\theta}$. n represents the total count of averaged samples.

The algorithm implements a two-phase buffer: odd-numbered extrema are stored temporarily in θ_{odd} . Subsequent even-numbered extrema trigger the update using both values of the extremal pair, where maximum and minimum compensate each other. This way, after two extrema or one period an estimation for the equilibrium exists, and with further periods, it becomes increasingly more accurate.

¹Due to noise and smoothing, $a = 0$ denotes the degenerate case of exactly zero velocity which is limited to floating-point precision artifacts.

Algorithm 2 Oscillation Detector (OCD)

<i>Parameters</i>		Default
n_P	Number of periods before convergence	2
<i>Variables</i>		Initial value
t	Time step counter	0
θ_{old}	Sensor value from last call	0
a	Direction sign (-1, 0, +1) of v	0
n_E	Number of detected extrema	0
θ_{odd}	State of last odd extremum	0
θ_{avg}	Average of detected extrema, est. equilibrium	0
UPDATE($\tilde{\theta}$)		
1:	if $t > 0$ then	
2:	$\Delta\theta = \tilde{\theta} - \theta_{\text{old}}$	
3:	if $\Delta\theta a < 0$ then	
4:	if $\text{mod}(n_E, 2) = 0$ then	▷ Extremum is odd
5:	$\theta_{\text{odd}} \leftarrow \tilde{\theta}$	
6:	else	▷ Extremum is even
7:	$\theta_{\text{avg}} \leftarrow \frac{(n_E-1)\theta_{\text{avg}} + \theta_{\text{odd}} + \tilde{\theta}}{n_E+1}$	
8:	end if	
9:	$n_E \leftarrow n_E + 1$	
10:	end if	
11:	$a \leftarrow \text{sign}(\Delta\theta)$	
12:	end if	
13:	$\theta_{\text{old}} \leftarrow \tilde{\theta}$	
14:	$t \leftarrow t + 1$	
DETECT()		
1:	return $(n_E \geq 2n_P)$	
EQUILIBRIUM()		
1:	return θ_{avg}	

The detection criterion is mathematically expressed as:

$$n_E \geq 2n_P,$$

where n_E is the number of so far detected extrema and n_P is the parameter that declares how many periods are completed before convergence. This method is insensitive to oscillation amplitude variations and tolerates irregular periodicity, as is commonly the case with c-mode controlled equilibria.

Figure 20 examines the same heteroclinic transition presented in Figure 19, with a focus on the left leg’s c-mode dynamics. Following the $(+, \cdot)$ -switch command, the c-mode controller drives positive angular displacement against ground contact forces within the left leg. Then at $t = 7.28\text{s}$, the overshoot beyond the true equilibrium becomes large enough to trigger a controller response in the opposite direction and marks the first extremum. Repeating this dynamic, the system settles into irregular saw-tooth-shaped oscillations with period $T \approx 8.2\text{s}$ (disregarding the irregularity) and amplitude $A \approx 0.8\text{rad}$. The gray line shows the $\theta_{\text{avg}}(t)$ estimation. The estimate after one full period $\theta_{\text{avg}}(t = 12.9\text{s}) = 5.02\text{rad}$ is already close to the equilibrium estimation after 5 periods $\theta_{\text{avg}}(t = 37.1\text{s}) = 4.98\text{rad}$.

5.2 Dead-zone

The irregularity in Figure 20 observed at $t = 25.7\text{s}$ exemplifies a typical dead-zone entrapment. For c-mode controlled joints, the absence of motion provides clear evidence of the motor getting stuck likely due to friction. Here, a previously described fixpoint detection identifies the convergence, and the controller responds with a voltage ramp to restore movement.

Although not originally anticipated as part of this work, dead-zone effects posed a critical practical challenge during the implementation of the ABC exploration algorithm, necessitating analysis and compensation. This section formalizes the dead-zone concept, elaborates its implications within the context of ABC exploration, and presents a gravity-aware compensation strategy.

5.2.1 Formal Definition

The dead-zone phenomenon arises from several physical mechanisms. At its core, static friction, or *stiction* for short establishes a force threshold that must

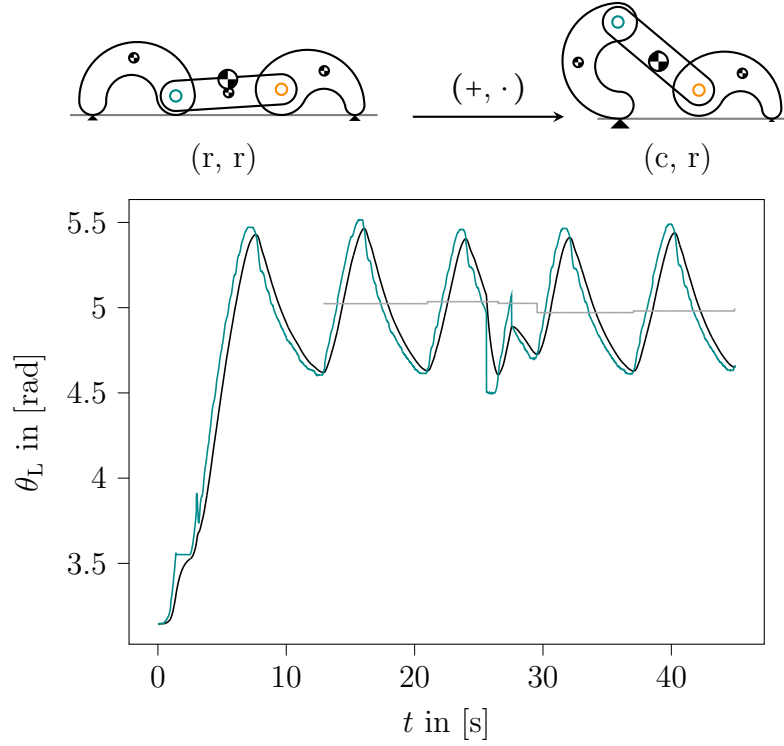


Figure 20: During the heteroclinic orbit transition $(r, r) \xrightarrow{(+, \cdot)} (c, r)$ seen here, the left leg angle θ_L is actively switched into c-mode and pulls the left joint off the ground by rotating positively. Instead of settling at an unstable fixpoint, a sawtooth-shaped oscillation arises due to overshoot and controller response. The cyan line shows the raw θ_L , the smoothed $\tilde{\theta}_L$ is given in black, and the grey line shows the development of θ_{avg} . After one period, an estimate $\theta_o \approx \theta_{avg} = 5.02 \text{ rad}$ exists. The irregularity at $t = 25.7 \text{ s}$ is an example of the motor getting stuck in a dead-zone. (sampling frequency $f_s = 100 \text{ Hz}$, smoothing factor $\alpha = 0.02$)

be exceeded to initiate relative motion between contacting bodies, such as between the leg and the ground or within a motor's bearing. Complementing this effect, mechanical backlash introduces temporary motion loss during direction reversals, particularly evident in gear trains where meshing clearances cause measurable delays when changing rotational direction.

The dead-zone $D(u)$ is a nonlinearity around the origin modeled mathe-

matically (slightly modified from [18]) as:

$$D(u) = \begin{cases} u - u_+, & u > u_+ \\ 0, & -u_- \leq u \leq u_+ \\ u + u_-, & u < -u_- \end{cases},$$

where u_+ and u_- represent positive and negative activation thresholds. In the context of DC motor control, this manifests as a nonlinear relationship between input voltage u and angular velocity $\dot{\theta}$:

$$\dot{\theta} = k \cdot D(u),$$

with k denoting the motor's velocity constant. The function is illustrated in Figure 21 (a) and features these characteristic regions:

1. an inactive region with a width of $w_D = u_+ + u_-$,
2. a critical input u_+ and u_- on either side, where the motor slips into motion
3. a *moving phase*, that displays the linear relationship between u and $\dot{\theta}$.

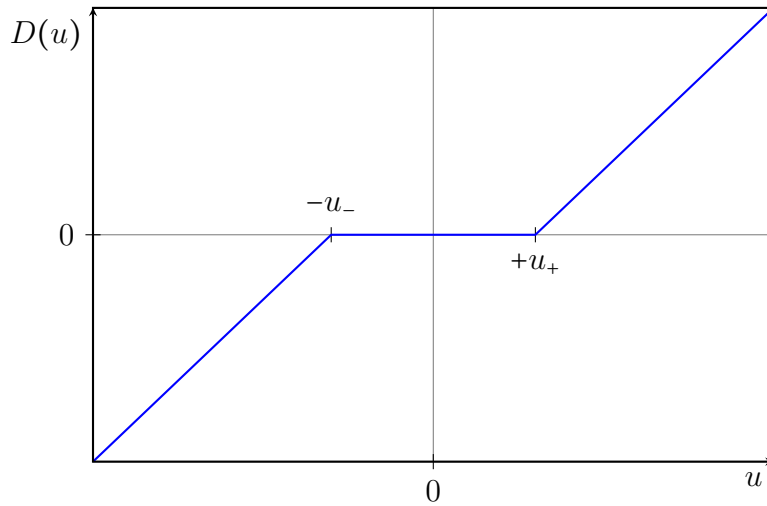


Figure 21: Mathematical model of the dead-zone

5.2.2 Dead-zone-induced Uncertainty

The ABC’s reliance on extrinsic forces for r-mode transitions creates a particular sensitivity to dead-zone effects. When joints under r-mode control (where $u = 0$) experience insufficient external torque, the joint angle gets *stuck* in states that the system may falsely identify as *pseudo*-stable equilibria.

Figure 22 demonstrates this phenomenon through an experimental motion sequence, contrasting theoretical predictions with observed behavior. (a) presents show the postures connected by switch mode transitions, and (b) shows the actual trajectory in the configuration space. Beginning at relaxed posture A, the application of the $(+, \cdot)$ -switch reliably drives the system to (c,r)-mode posture B, where the right joint gets passively rotated due to ground contact. However, application of the inverse $(-, \cdot)$ -switch applied to the B posture fails to return the system to posture A. Due to dead-zone, the right joint preserves the positive angular offset, and the system reaches the pseudo-stable posture S.

The pseudo-stable states arising from dead-zones introduce uncertainties in the sensorimotor exploration process. These dead-zone states like S, are susceptible to disruption by minor environmental perturbations that overcome the dead-zone forces and cause slippage. The therefore stochastic nature of these pseudo-stable states leads to unreliable transition outcomes. Repetitions of this experiment yield varying final pseudo-stable postures across different trials. Identical switch actions $\mathbf{d}$ from the same source states $\mathbf{s}_i$ produce divergent destination states. Consequently, $\mathbb{G}$ can no longer deterministically represent the graph edges as $\mathbf{s}_i \xrightarrow{\mathbf{d}} \mathbf{s}_j$, and more complex probabilistic representations become necessary.

5.2.3 Dead-zone Asymmetry

During deployment, the motors rotational axes are parallel to the ground so that gravitational forces introduce significant direction-dependent biases and asymmetries to the dead-zone profile: when gravitational torque aligns with the intended direction, the critical threshold decreases (sometimes even going negative). For gravity-opposed rotations, it increases.

The dead-zone profiles are analyzed through dead-zone diagrams as in Figure 23. The diagrams examine the relationship between applied motor voltage

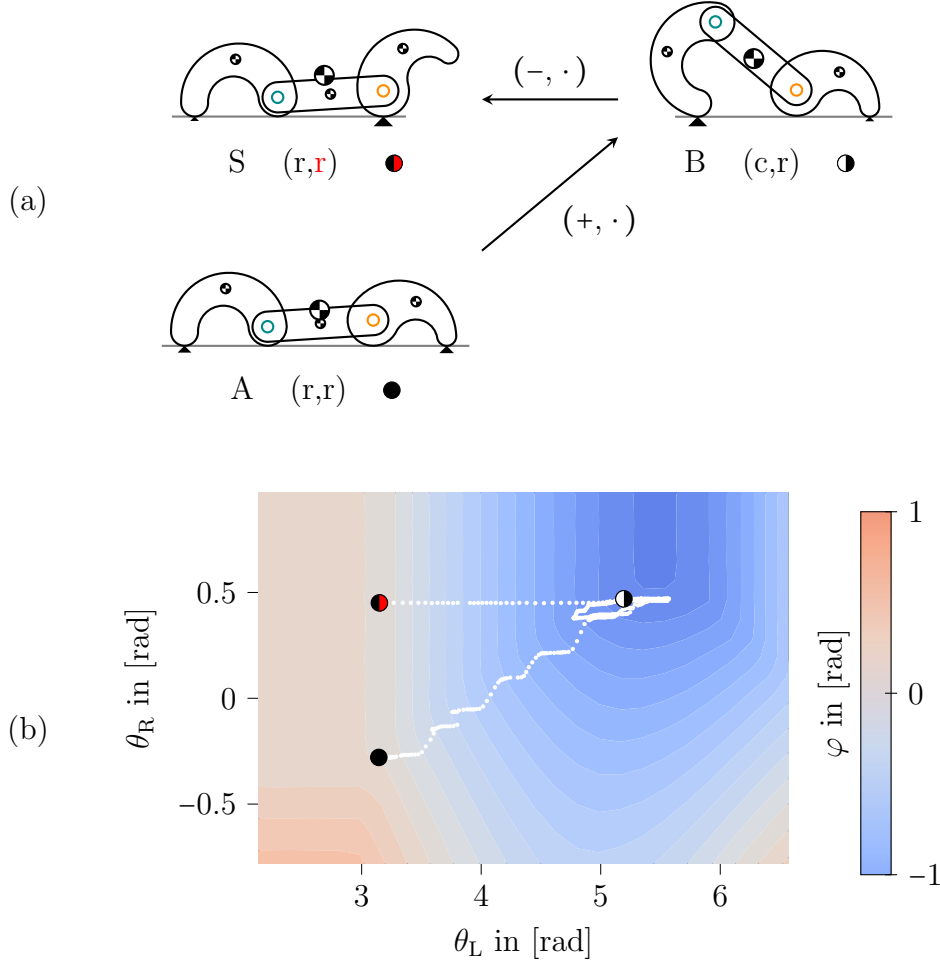


Figure 22: The transition $A \xrightarrow{(+, \cdot)} B \xrightarrow{(-, \cdot)} S$ illustrates how dead-zones introduce uncertainties to the ABC exploration. In (a), the featured states are schematized as postures, and (b) shows the trajectory (in white) in the configuration space. The underlying colored contour lines represent φ , computed in simulation. In theory, the $(-, \cdot)$ -switch should transition from B to A, but because the right leg is stuck, it does not fall back to the ground, and the system terminates in the pseudo-stable posture S instead.

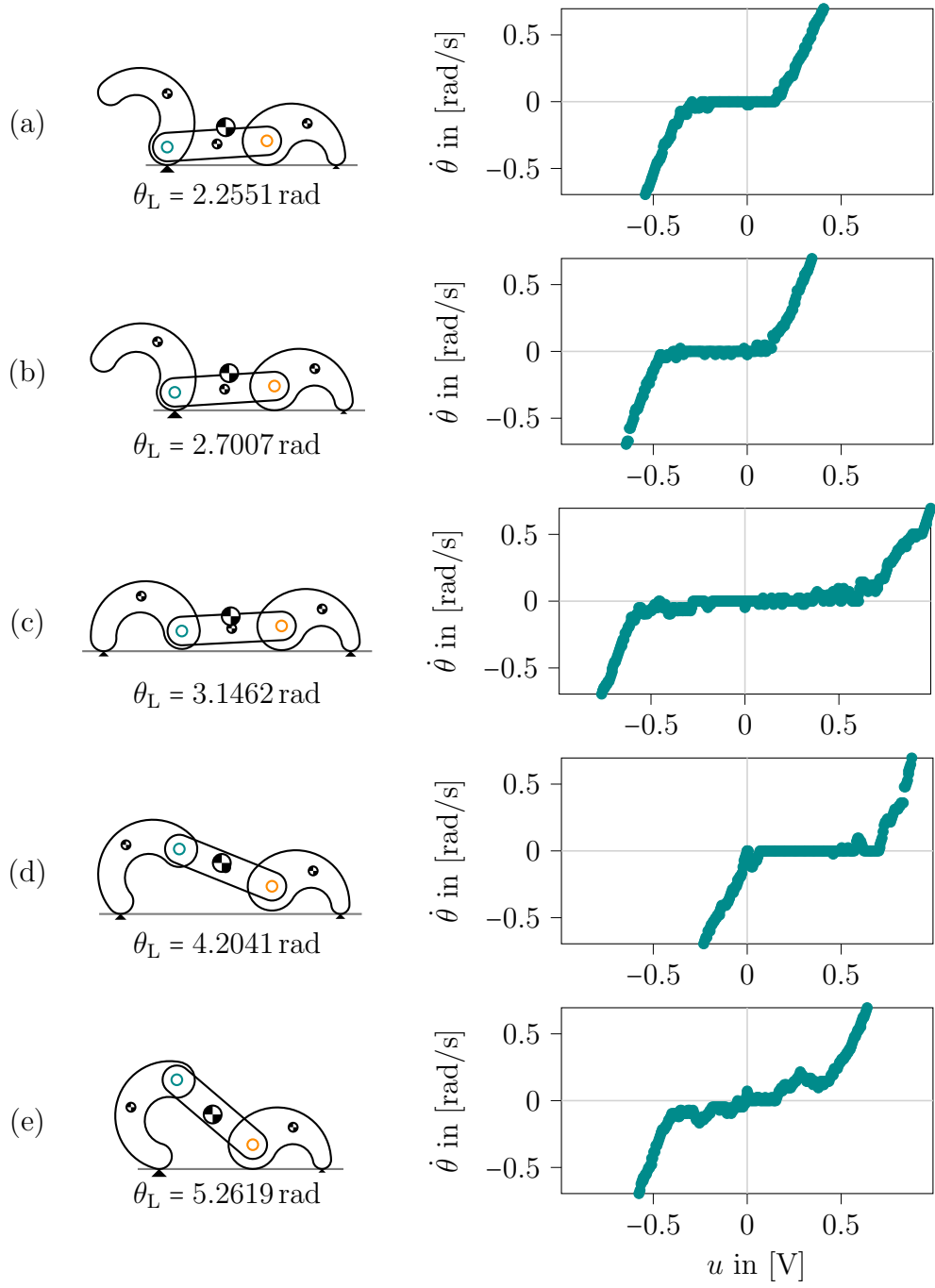


Figure 23: Asymmetric dead-zone diagrams for the θ_L -joint sampled at five different angles. When the motion direction is supported by gravity the dead-zone's critical threshold decreases, and motion that is opposed by gravity has an increased threshold.

$u(t)$ and resulting angular velocity $\dot{\theta}(t)$. The data points are collected by applying a slow voltage ramp (rate: 0.3 V/s) to the motor in either direction, originating from a state of rest. A video recording of the corresponding experiment is available on YouTube². The figure reveals five dead-zone profiles for the θ_L -joint, each differently influenced by gravity:

- (a) With the limb elevated to an unstable equilibrium, the dead-zone remains almost symmetric with close to no gravitational biases, as its orthogonal force components on the limb are negligible.
- (b) At the intermediate posture between the unstable (a) and the stable fixpoint (c), gravity pulls on the mass of the left leg in the positive direction. This creates an asymmetric deadband shifted into the negative direction with u_- increased and u_+ decreased.
- (c) This is the stable fixpoint posture A from Figure 22. As the left leg touches the ground, u_+ suddenly increases as the positive rotation must now overcome the combined mass of both leg and torso to lift them off the ground. u_- also reaches its peak because at this angle gravity pulls orthogonally on the legs weight.
- (d) The intermediate posture between (c) and (e) mirrors the asymmetry in (b) but with opposite alignment.
- (e) This is the unstable fixpoint pose B from Figure 22. Similar to (a) but with the limb positioned beneath the torso. In this state the dead-zone is again almost symmetric.

These experiments provide several key insights:

1. In any non-equilibrium configuration, the critical input voltage is asymmetric. Due to gravity, movement toward the stable equilibrium requires less input than moving toward the unstable equilibrium.
2. Progressively approaching the stable fixpoint further decreases the critical input threshold.
3. When moving toward the stable fixpoint, a sudden threshold increase emerges upon attaining it.

²<https://youtu.be/rwmssoSQIzQU>

5.2.4 Dead-zone Control

This work presents a novel control strategy for r-mode controlled joints suffering from dead-zones. The method exploits the dead-zone’s asymmetry to distinguish pseudo-stable from true equilibrium states and guides the controller toward stable convergence.

The Dead-zone Control algorithm (DZC) specified in Algorithm 3 for initialization and for iterative updates implements a three-phase state machine (denoted as “phases” to avoid confusion with “states” of dynamical systems):

- ramp** The voltage increases linearly according to the slope parameter Δu until the output velocity exceeds $v_{\max}$, indicating dead-zone escape and triggering the activation of the wait-phase.
- wait** Motors are disabled until the Fixpoint Detector confirms angular stability. Upon detection, the algorithm assumes pseudo-stability, and the ramp phase activates in the opposite direction: $d \leftarrow \neg d$ (where $\neg$ inverts the direction).
- idle** The terminal phase indicating true stable convergence where motors remain disabled as in conventional r-mode control.

The algorithm distinguishes pseudo-stable from true stable states by comparing current power consumption with previously saved power thresholds p_+ and p_- . When velocity exceeds $v_{\max}$ in a certain direction $d \in \{+, -\}$, the corresponding directional threshold $p_{(d)}$ updates:

$$\text{if } |\dot{\theta}| > v_{\max} \quad \text{then} \quad p_{(d)} \leftarrow u \cdot i.$$

True stable states are identified when current power exceeds the stored threshold ($p > p_{(d)}$) triggering the idle phase. Conversely, if $p > p_{(-d)}$ —the power threshold of the opposite direction—the algorithm infers that the true stable fixpoint lies opposite and aborts the ramp prematurely. This directional power comparison effectively exploits dead-zone asymmetries for discrimination of true and pseudo-stable states.

The presented control algorithm focuses on the core novel components necessary to demonstrate the proposed dead-zone compensation. Additional functionality would be required to handle stall detection and symmetry-breaking mechanisms. While essential for practical implementation, these extensions

have been omitted here to maintain clarity in presenting the DZC's core strategy.

Algorithm 3 Dead-zone Control (DZC)

<i>Parameters</i>		Default
Δu	Ramp voltage increment	0.5 V/s
$v_{\max}$	Slip target velocity	0.9 rad/s
<i>Variables</i>		Initial Value
u	Motor voltage	0
d	Ramp direction $\in \{+, -\}$	+
p_+	Positive power threshold	∞
p_-	Negative power threshold	∞
ACTIVATE()		
ramp		wait
1: $u \leftarrow 0$		1: FPD.Init()
UPDATE($\tilde{\theta}, \dot{\theta}, i$)		
ramp		wait
1: $u \leftarrow u + d\Delta u$		1: FPD.Update($\tilde{\theta}$)
2: $p = ui$		2: if FPD.DETECTED() then
3: if $ p > p_{(d)}$ then		3: $d \leftarrow \neg d$
4: ACTIVATE(idle)		4: ACTIVATE(ramp)
5: else if $ p > p_{(-d)}$ then		5: end if
6: ACTIVATE(wait)		6: MOTOR.DISABLE()
7: end if		
8: if $ \dot{\theta} > v_{\max}$ then		
9: $p_{(d)} \leftarrow p $		
10: ACTIVATE(wait)		
11: end if		
12: MOTOR.ENABLE()		
13: MOTOR.VOLTAGE $\leftarrow u$		

5.3 Exploration Results

This subsection presents the experimental results of the ABC exploration, demonstrating the algorithm’s performance on the physical robotic platform. Figure 24 shows the four immediate heteroclinic cycles surrounding the (r,r)-mode posture E, where both legs are spread away from the torso limb. The exploration algorithm successfully identified these adjacent cycles within close to 7 min of operation. The video recording of this exploration can be found on YouTube³.

The schematic representation in (a) aligns the robot’s postures with their corresponding states in the topological graph structure, while (b) plots the actual trajectory through configuration space, with θ_L and θ_R represented on the x- and y-axes respectively. The contour lines indicate the simulated body orientation angle φ . Red markers highlight pseudo-stable postures encountered during exploration.

Notably, the transition $F \xrightarrow{(-,\cdot)} E$ demonstrates the dead-zone compensation (DZC) successfully overcoming the uncertainty shown in Figure 22’s transition $B \xrightarrow{(-,\cdot)} S$. Similarly, the transition $H \xrightarrow{(\cdot,+)} E$ shows the DZC guiding the left leg from a pseudo-stable state to the true equilibrium in E.

Topological Insight The boundaries where distinct smooth manifold patches intersect, i. e., discontinuities in the gradient $\nabla\varphi(\theta_L, \theta_R)$ manifest geometrically as creases on the $\mathbb{M}_\theta(\varphi)$ -manifold’s surface. Crucially, this experiment reveals that equilibrium states naturally align with these crease features, exemplified by all heteroclinic orbits connecting to posture $E \leftrightarrow \{B, D, F, \text{ or } H\}$. As observed in Section 3.3.2 (Sensorimotor Manifolds), these discontinuities correspond to postures where a GCP is either introduced to or removed from the ground surface. Notably, these contact-switching events coincide with r-mode controlled joints. Through the topological structure of equilibrium states in the exploration graph alone, a change in $\nabla\varphi(\theta_L, \theta_R)$ can be inferred without directly measuring φ .

³<https://youtu.be/uti2gKSJXo4>

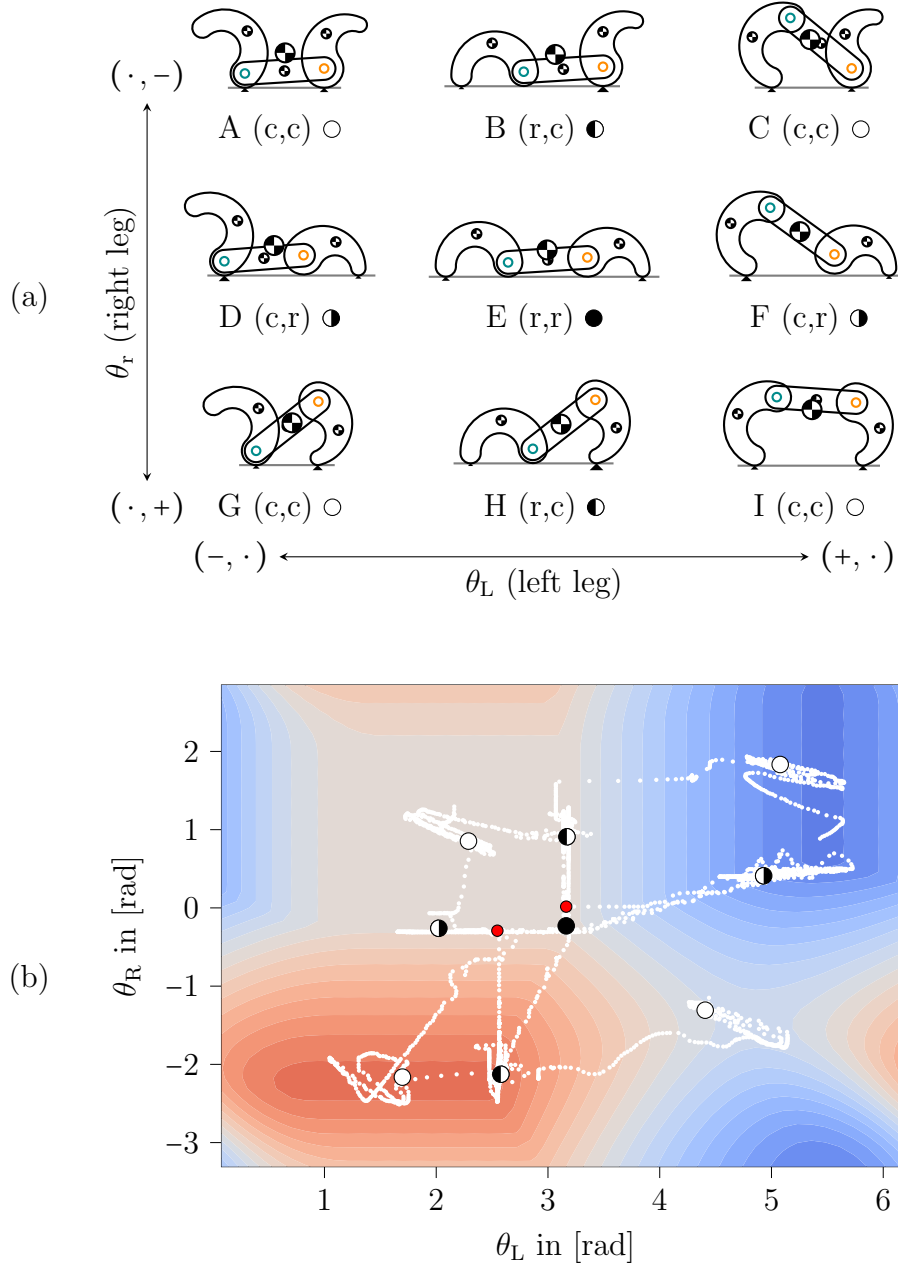


Figure 24: The four immediate core cycles around the initial relaxed posture E. The schematic postures in (a) match the marked equilibria in (b) which shows the travelled trajectory during exploration in the configuration space. The contour lines represent the body angle φ from simulation.

6 Locomotion

This section demonstrates a practical application of the sensorimotor manifold, or rather a discrete approximation thereof. Building upon the exploration graph $\mathbb{G}$ developed in the previous sections, I examine how topological properties of the manifold enable motion generation. Firstly, a sliding-style gait is defined through geometric characteristics of key postures. The second subsection uses only the topological structure of the explore graph to infer candidate states for locomotion. Finally, a planning strategy is designed and presented that translates these topological features into controlled movement.

6.1 Sliding-style Locomotion

This section gives a brief description of the locomotion style that is considered in this thesis: namely, a *statically stable* sliding gait, where “a statically stable gait is periodic locomotion in which the biped’s COM does not leave the support polygon” [3]. In the planar robot case, the support polygon refers to the line segment between the outermost GCPs. This gait operates entirely within the sensorimotor manifold, i. e., maintaining minimal angular velocity ($\dot{\varphi}$) and acceleration ($\ddot{\varphi}$) throughout the motion. This approach differs from dynamic methods like toppling or running and instead relies on controlled weight shifting between contact points.

For this, I introduce two key geometric properties of postures: the distance between the GCPs, as

$$\Delta p = p_R - p_L,$$

and the normalized weight distribution

$$w = 2 \frac{c_x - p_L}{\Delta p} - 1,$$

where $w = -1$ indicates complete weight on the left contact point (p_L) and $w = 1$ on the right (p_R).

By varying Δp and w appropriately, the three key postures in Figure 25 are periodically cycled through to achieve forward motion from left to right:

- A The cycle starts from a balanced configuration ($w \approx 0$) with minimal Δp .
- A→B Extending Δp while biasing weight backward ($w < 0$) enables forward sliding of the front leg (right).

B→C The weight is shifted forward ($w > 0$) while keeping Δp extended.

C→A To close the cycle, the robot contracts Δp . The forward weight bias ($w > 0$) causes the hind leg (left) to slide producing net displacement. The robot finds itself back in the original posture, ready to repeat.

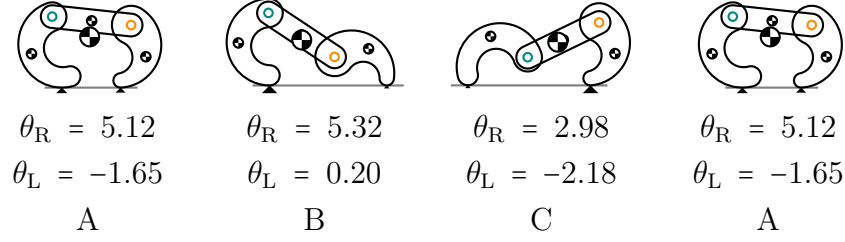


Figure 25: The exemplary cycle of key postures illustrates the sliding-style gait (postures taken from [51]). By positioning its weight closer to one side (w), the robot can control which leg will stick and which will slip when varying the distance between the GCP (Δp).

The cycle in Figure 25 given was generated in previous work[51], by hand-tuning target values for Δp and w for each key posture and then locating optimal states within the simulated SMM. While simulation enables direct computation of Δp and w , the physical robot faces limitations in acquiring these properties without additional sensors or computationally expensive physics engines.

6.2 Topologically Inferring Locomotion

This section develops a topological approach to infer locomotion-relevant properties from the exploration graph structure.

Given Brilliance’s constraint to its submanifold (see Section 4.2), I refine the heteroclinic orbit notation to emphasize bidirectionality:

$$\mathbf{s}_{*i} \xleftrightarrow{\mathbf{d}} \mathbf{s}_{*j}.$$

where $\mathbf{d} = (d_L, d_R)$ follows the established notation with $(\cdot)$ for passive joints. Active switching on a bidirectional edge can be indicated with $\pm$ when both $+$ and $-$ are possible.

A *core cycle* $\mathcal{C}$ refers to a connected subgraph of the exploration graph $\mathbb{G}$ containing one instance of each fundamental equilibrium type:

$$\{A_{\circ} \leftrightarrow B_{\bullet} \leftrightarrow D_{\bullet} \leftrightarrow C_{\bullet} \leftrightarrow A_{\circ}\} = \mathcal{C} \subset \mathbb{G}.$$

Figure 26 isolates the core cycle most suitable for locomotion from the graph gained in the previous section. (a) is displaying the corresponding physical postures, and (b) shows the four states in the configuration space (superimposed with the hand-tuned cycle in white for reference).

The sliding locomotion mechanism critically depends on proper ground contact of both leg limbs. As established in Section 5.3, this condition can be inferred through topological analysis of $\mathbb{G}$. Passive r-mode joint displacement during heteroclinic transitions can occur from two scenarios:

1. Indirect ground-mediated displacement, where body reorientation (φ) induces motion from the active joint to the passive one. Here, the displacement serves as a topological proxy for ground contact.
2. Direct limb-to-limb contact, where the active limb mechanically pushes the passive one. This is undesirable for locomotion but can be identified through stall edges and then excluded from consideration.

To quantify the compliance behavior of passive joints during state transition, I introduce the *passive angular displacement* metric $\Delta\theta_{(\cdot)}$ for heteroclinic transitions:

$$\Delta\theta_{(\cdot)}(\theta_{*i} \xleftrightarrow{(d_L, d_R)} \theta_{*j}) = \begin{cases} |\theta_{Li} - \theta_{Lj}|, & d_L = (\cdot) \\ |\theta_{Ri} - \theta_{Rj}|, & d_R = (\cdot) \end{cases}$$

which measures the absolute angular difference of the passive joint angle during a transition.

Figure 26 holds annotation for the passive angular displacements during transitions originating from the relaxed $D_{\bullet}$ posture. The edge $D_{\bullet} \leftrightarrow B_{\bullet}$ demonstrates the r-mode controlled right limb undergoing a passive displacement of 0.6558 rad as the actively controlled left limb elevates itself and the torso limb. Conversely, the transition $D_{\bullet} \leftrightarrow C_{\bullet}$ shows the symmetric case with passive left-leg compliance and a displacement of 0.5975 rad.

Release-mode joints maintaining ground contact reside precisely at creases in the sensorimotor manifold (SMM)—the areas where smoothness breaks due

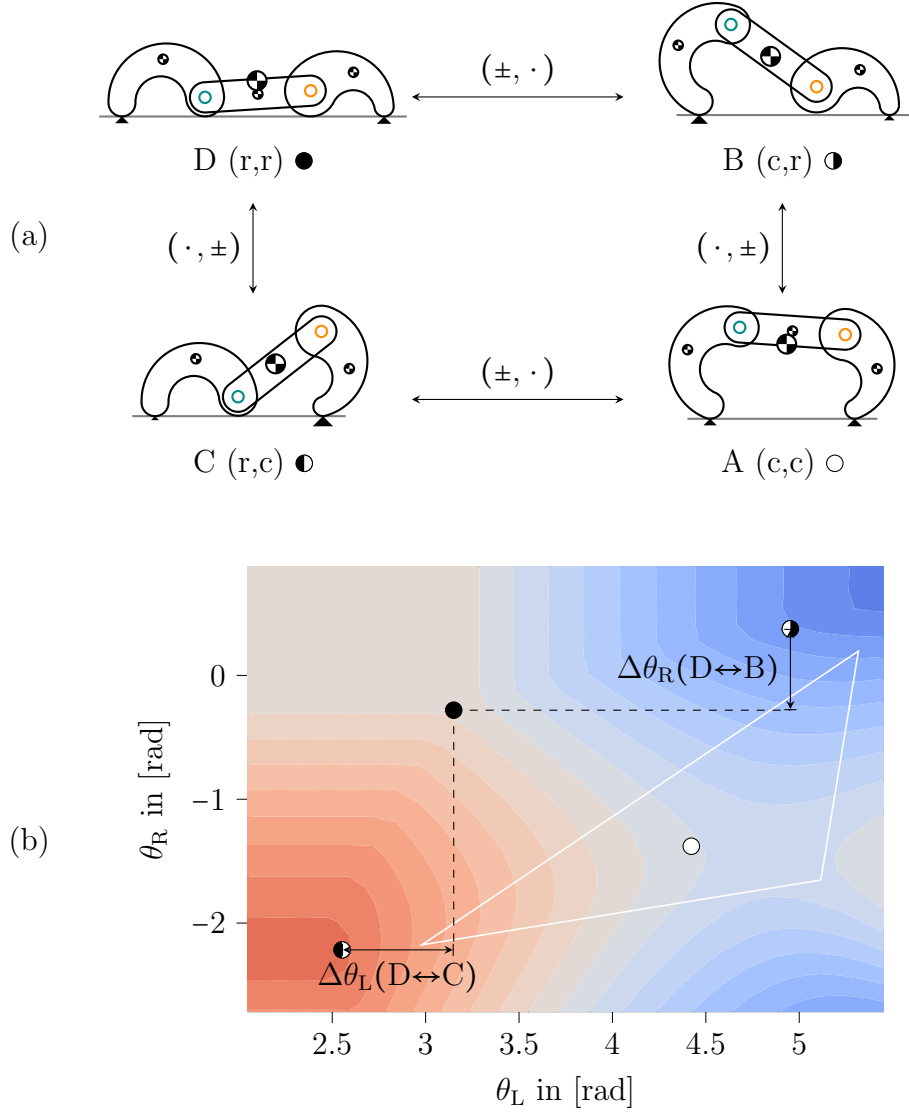


Figure 26: The presented core cycle from Brilliance’s exploration is most relevant for locomotion. (a) gives the corresponding postures and bidirectional transitions. (b) shows the equilibria in configuration space, the hand-tuned gait superimposed in white, and annotations of the passive r-mode angular distance $\Delta\theta_{(\cdot)}$. For the $D \leftrightarrow B$ -transition, the right leg is passive, while the left leg is passive for $D \leftrightarrow C$.

to changes in ground contact (GCP). Joints situated at these critical positions maintain ground contact when complying with passive displacement, whereas exaggeration of the passive displacement would result in immediate lift-off of the limb.

This leads to an assumption: a GCP that is ready to be lifted off the ground is unlikely to carry much weight. The saddle point postures $B_\bullet$ and $C_\bullet$ exhibit this asymmetric weight distribution ($w \neq 0$) by preferentially loading the GCP of the c-mode controlled joint.

For a core cycle $\mathcal{C} \subset \mathbb{G}$ exhibiting substantial passive displacements $\Delta\theta_{(\cdot)}$ in both $D_\bullet \leftrightarrow B_\bullet$ and $D_\bullet \leftrightarrow C_\bullet$ transitions, as is the case in Figure 26, an intermediate (c,c)-mode state $A_\circ$ exists. Both limbs maintain ground contact while the torso limb is lifted. In this state, the c-mode-controlled joints actively regulate their positions against ground contact forces, likely converging to a contracted GCP distance Δp compared to saddle point postures. The c-mode control maintains balanced weight distribution ($w \approx 0$).

The identified equilibrium states— $A_\circ$, $B_\bullet$, and $C_\bullet$ —embody the key characteristics required for the sliding-style gait.

6.3 Planning

The topological analysis enables a formal motion planning recipe that synthesizes a three-posture sliding gait derived from the exploration graph $\mathbb{G}$. Algorithm 4 implements this planning paradigm. To locate suitable core cycles, a scoring function is introduced:

$$\text{score}(\mathcal{C}) = \min \left(\Delta\theta_{(\cdot)}(D_\bullet \leftrightarrow B_\bullet), \Delta\theta_{(\cdot)}(D_\bullet \leftrightarrow C_\bullet) \right).$$

It quantifies the minimum of both passive r-mode displacements within the cycle. The cycle $\mathcal{C}^*$ in $\mathbb{G}$ with the highest score provides the key postures $A_\circ$, $B_\bullet$, and $C_\bullet$ corresponding to the unstable fixpoint and two saddle points respectively.

Practical implementation requires slight angular adjustments (ϵ) of the saddle state's r-mode angle toward the unstable posture $A_\circ$ to guarantee they maintain proper ground contact and to ensure reliable weight transfer during locomotion execution.

Algorithm 4 Locomotion Planner

<i>Parameters</i>		Default
ϵ	Ground contact adjustment	0.2 rad
PLAN($\mathbb{G}$)		
1: $\mathcal{C}^* = \operatorname{argmax}_{\mathcal{C} \in \mathbb{G}} [\operatorname{score}(\mathcal{C})]$		
2: $A = \circ(\mathcal{C}^*)$		
3: $B = \bullet(\mathcal{C}^*)$		
4: $C = \bullet(\mathcal{C}^*)$		
5: $\theta_R(B) \leftarrow \theta_R(B) + \epsilon d_R$, with $B \xrightarrow{(\cdot, d_R)} A$		
6: $\theta_L(C) \leftarrow \theta_L(C) + \epsilon d_L$, with $C \xrightarrow{(d_L, \cdot)} A$		
7: return (A, B, C)		

6.4 Locomotion Results

The implemented planning framework successfully generates effective locomotion sequences for the Brilliance morphology with the complete angular trajectory specified in Table 5. The motion profile is created by interpolating between these angles using 5 s transitions controlled through a standard PI-controller.

Table 5: Brilliance’s locomotion sequence

Pose	θ_L [rad]	θ_R [rad]	$\approx \varphi$ [rad]
A	4.810 51	−1.649 88	−0.102 35
B	2.731 64	−2.084 21	−0.551 50
C	5.261 94	0.275 29	0.566 22

Figure 27 presents the resulting motion: (a) the key postures at the corners of the trajectory triangle and (b) the configuration space trajectory. Experimental recordings of the locomotion are uploaded to YouTube⁴.

The energy analysis in (c) reveals the power consumption across the locomotion sequence for the left (p_L) and right (p_R) sides:

A $\rightarrow$ B The transition requires minimal energy for either joint as gravity assists the right leg’s outward folding and the left leg barely moves.

⁴<https://youtu.be/aeyrxt5Uz3w>

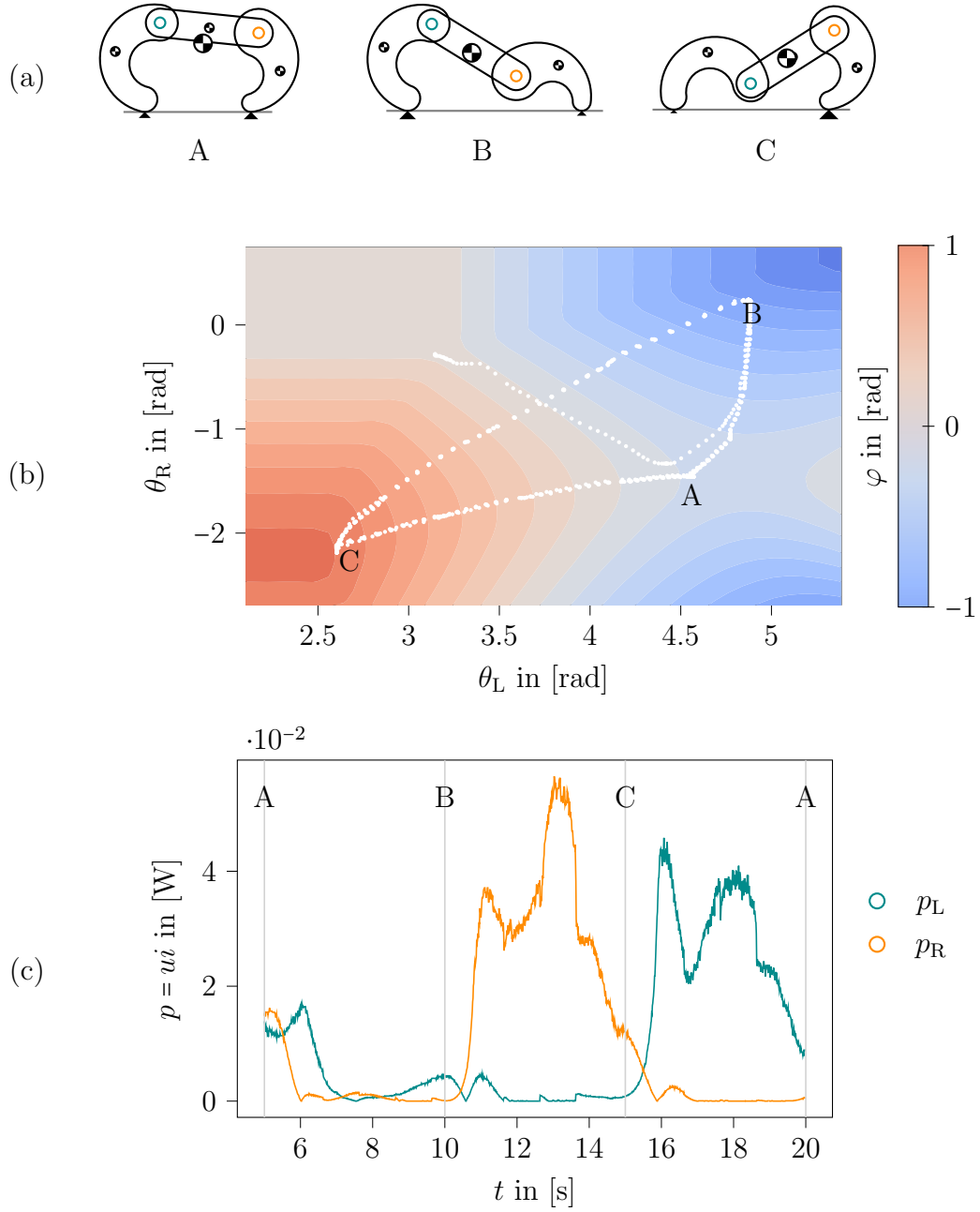


Figure 27: Visualization of the synthesized locomotion sequence for the Brilliance morphology. (a) Shows the gait's key postures at the corners of the trajectory triangle, (b) gives the planned trajectory in configuration space, and (c) displays the power consumption $p = ui$ for each joint during the execution of the sequence. The left power p_{textL} is given in cyan and the right p_{textR} in orange.

B $\rightarrow$ C Here, the left leg's gravity-assisted motion contrasts with the right leg's active ground-pushing, causing energy expenditure at p_R .

C $\rightarrow$ A Analogously, the closing transition produces a p_L -peak as the left leg actively elevates against gravity while the right only stabilizes.

Integrating across five cycles produces a total average energy expenditure of

$$E = \int_{A \odot} (p_L(t) + p_R(t)) dt = 0.2924 \text{ J}$$

per cycle, where A $\odot$ denotes the time interval of one cycle starting and ending at pose A. The average movement speed achieved is approximately 10 cm per cycle, so 0.67 cm/s.

7 Conclusion

This concluding section is organized into three parts: firstly, a summary of the entire thesis’s concepts and results; secondly, a critical analysis of the work’s constraints and underlying necessary assumptions; and finally, a perspective on potential extensions that could advance the proposed methodology outlining promising directions for future research.

7.1 Summary

In this thesis, I have developed a framework for generating locomotive behaviors by leveraging the topological structure derived from attractor-based exploration. The core innovation lies in utilizing the topological properties of sensorimotor manifolds (SMM)—the stable state configurations that emerge when viewing robotic systems through a dynamical systems perspective. The topology of these manifolds reflects geometric properties of the morphology embedded in the physical environment, hence providing a powerful representation for motion planning.

The approach builds upon Cognitive Sensorimotor Loops (CSL) that locate equilibria from individual joints through two control modes: stable fixpoints are found in release-mode (r-mode) by complying with outside forces, and contraction-mode (c-mode) opposes outside forces to converge toward unstable fixpoints. The Attractor-Based Behavior Control (ABC) algorithm systematically explores the state space by switching CSL modes to induce heteroclinic orbits as connections between equilibria. The resulting graph structure serves as a sparse discretization of the underlying SMM, enabling its practical use for graph algorithms and robotic applications.

A key contribution addresses the challenges posed by motor dead-zone nonlinearities during physical implementation. Especially during r-mode control, the dead-zone phenomenon introduces uncertainties into the graph structure by getting stuck in pseudo-stable equilibria rather than converging to the true stable ones. The proposed method exploits dead-zone asymmetries to reliably distinguish true stable fixpoints from pseudo-stable states, improving the robustness of the ABC framework without requiring explicit uncertainty modeling.

To answer the research question posed in the Section 1 (Introduction), this thesis demonstrates how a planar, three-limb robot can derive a locomotion behavior from its self-explored ABC-graph. Passive angular displacement in r-mode-controlled joints during transitions is indicative of postures with locomotion-relevant ground contact conditions. The method searches the graph for core cycles that feature this angular displacement and supply the key postures for a sliding-style locomotion.

The presented approaches have both been tested and validated on the physical robot Brilliance. The ABC exploration extended by the dead-zone control has enabled the robot to successfully self-explore its own sensorimotor manifold free of dead-zone-induced uncertainties. The purely topology-driven planning generated a gait consisting of three key postures. The robot was able to move forward with this gait, demonstrating the effectiveness of the suggested strategy.

7.2 Limitations

The current framework incorporates several simplifying constraints to investigate fundamental principles while at the same time avoiding the curse of dimensionality. The approach is restricted to planar morphologies with rotational axes orthogonal to the sagittal plane, operating on flat, obstacle-free terrain, and employing just three limbs with two actuated joints.

The locomotion planning approach has been validated specifically on the Brilliance morphology serving primarily as a proof of concept. This validation rests on the key assumption that the exploration graph contains suitable core cycles exhibiting passive r-mode displacements for both joints—a condition that may not hold for all morphological designs, particularly those with significantly disproportionate limb sizes.

Broader evaluation within the scope of this thesis was constrained by time limitations for physical experimentation and the combinatorial explosion of possible morphological variations (shape parameters, variation in joint and center of mass positions, different limb materials, etc.).

7.3 Outlook

The ABC framework could be extended through analysis of unstable fixpoint oscillations that emerge during c-mode control. This includes studying the relationship of CSL parameters (g_i, g_f) on oscillation period and amplitude or using the CSL internal state to detect convergence. Incorporating body velocity $(\dot{\varphi})$ for momentum-based motion generation allows for dynamic exploration and motion planning.

Morphological studies could investigate how shape, mass distribution, and material properties affect locomotion. Energy-optimized motion planning and more abstract behavior representations could extend the current key-posture approach. Future work should also explore higher-dimensional systems. The ABC approach shows potential to tame the curse of dimensionality because complexity only grows linearly with increasing DOF. To benchmark the approach’s scalability and performance it can be compared against popular locomotion planning methods like reinforcement learning or artificial evolution.

The topological insights gained from studying SMM could eventually be used to guide robot engineers to make informed design choices to explicitly model for certain behavioral capabilities. A promising long-term direction involves architectures consisting of exploration graphs at a lower level building up to high-level memory structures. The ultimate goal is still to create truly autonomous systems that can explore themselves and behave in ways that are adaptive, as Brooks and Asimov envisioned. To view a machine as its own intelligent being it has to be able to gather information and derive behavior without explicit programming. The pursuit lies here not only in creating a versatile addition to a human’s daily life but also in gaining a more profound understanding of what intelligence truly is. So that one day, we can understand why Andrew has this streak of artistic ability in him.

List of Figures

1	Examples of manifolds	11
2	Homeomorphic relationship of the $\mathbb{A}^2$ and the $\mathbb{T}^2$	12
3	Flowcharts of dynamical systems	13
4	Vector fields of four fundamental types of 2D equilibria	15
5	The four different kinds of attractors	18
6	Illustration of a basin and its separatrices	19
7	Brilliance morphology at selected postures	21
8	Brilliance's 1D state space and equilibrium postures	23
9	Brilliance's 2D bifurcation diagram	25
10	Brilliance's 3D sensorimotor manifold	27
11	CSL flowchart	28
12	Toy attractor landscape	31
13	Lattice structure of the ABC switch modes	35
14	Brilliance: photo and schematic	37
15	Hardware and electronic components	39
16	The design pipeline of a limb (right leg)	41
17	Submanifold, 3D and 2D projection	42
18	Explorer update flowchart	44
19	FPD: transition to stable fixpoint	47
20	OCD: transition to oscillation around unstable fixpoint	51
21	Dead-zone: mathematical function	52
22	Dead-zone: uncertainties in the ABC graph	54
23	Dead-zone: asymmetric dead-zone diagrams	55
24	The four explored core cycles	60
25	Exemplary gait of key posture	62
26	Isolated locomotion-relevant core cycle	64
27	Brilliance's synthesized locomotion sequence	67

List of Tables

1	CSL-modes parameters	28
2	ABC state modes	32
3	Motor specifications	38
4	Parameters of the Brilliance morphology	41
5	Brilliance's locomotion sequence	66

List of Algorithms

1	Fixpoint Detector (FPD)	46
2	Oscillation Detector (OCD)	49
3	Dead-zone Control (DZC)	58
4	Locomotion Planner	66

References

- [1] I. Asimov and R. Silverberg, *The positronic man*. Doubleday Books, 1992.
- [2] R. A. Brooks, “Intelligence without representation”, *Artificial intelligence*, vol. 47, no. 1-3, pp. 139–159, 1991.
- [3] E. R. Westervelt, J. W. Grizzle, C. Chevallereau, J. H. Choi, and B. Morris, *Feedback control of dynamic bipedal robot locomotion*. CRC press, 2018.
- [4] M. Hildebrand, “Symmetrical gaits of horses: Gaits can be expressed numerically and analyzed graphically to reveal their nature and relationships.”, *Science*, vol. 150, no. 3697, pp. 701–708, 1965.
- [5] P. Biswal and P. K. Mohanty, “Development of quadruped walking robots: A review”, *Ain Shams Engineering Journal*, vol. 12, no. 2, pp. 2017–2031, 2021.
- [6] J. Coelho, F. Ribeiro, B. Dias, G. Lopes, and P. Flores, “Trends in the control of hexapod robots: A survey”, *Robotics*, vol. 10, no. 3, p. 100, 2021.
- [7] M. H. Raibert, “Hopping in legged systems—modeling and simulation for the two-dimensional one-legged case”, *IEEE Transactions on Systems, Man, and Cybernetics*, no. 3, pp. 451–463, 2012.
- [8] T. Mikolajczyk et al., “Recent advances in bipedal walking robots: Review of gait, drive, sensors and control systems”, *Sensors*, vol. 22, no. 12, p. 4440, 2022.
- [9] G. S. Chirikjian and J. W. Burdick, “The kinematics of hyper-redundant robot locomotion”, *IEEE transactions on robotics and automation*, vol. 11, no. 6, pp. 781–793, 2002.
- [10] M. Calisti, G. Picardi, and C. Laschi, “Fundamentals of soft robot locomotion”, *Journal of The Royal Society Interface*, vol. 14, no. 130, p. 20170101, 2017.
- [11] M. Yim, “New locomotion gaits”, in *Proceedings of the 1994 IEEE International conference on Robotics and Automation*, IEEE, 1994, pp. 2508–2514.

- [12] T. McGeer, “Principles of walking and running.”, *Mechanics of Animal Locomotion.*, 1992.
- [13] S. Collins, A. Ruina, R. Tedrake, and M. Wisse, “Efficient bipedal robots based on passive-dynamic walkers”, *Science*, vol. 307, no. 5712, pp. 1082–1085, 2005.
- [14] M. Jelali and B. Huang, “Detection and diagnosis of stiction in control loops: State of the art and advanced methods”, 2009.
- [15] H. Olsson, K. J. Åström, C. C. De Wit, M. Gäfvert, and P. Lischinsky, “Friction models and friction compensation”, *Eur. J. Control*, vol. 4, no. 3, pp. 176–195, 1998.
- [16] G. Zames and N. Shneydor, “Dither in nonlinear systems”, *IEEE Transactions on Automatic Control*, vol. 21, no. 5, pp. 660–667, 2003.
- [17] T. Hägglund, “A friction compensator for pneumatic control valves”, *Journal of process control*, vol. 12, no. 8, pp. 897–904, 2002.
- [18] G. Tao and P. V. Kokotovic, “Adaptive control of plants with unknown dead-zones”, *IEEE transactions on automatic control*, vol. 39, no. 1, pp. 59–68, 1994.
- [19] Q. H. Xia, S. Y. Lim, M. H. Ang, and T. M. Lim, “Adaptive joint friction compensation using a model-based operational space velocity observer”, in *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA’04. 2004*, IEEE, vol. 3, 2004, pp. 3081–3086.
- [20] J.-Y. Jeon, J.-H. Kim, and K. Koh, “Experimental evolutionary programming-based high-precision control”, *IEEE Control Systems Magazine*, vol. 17, no. 2, pp. 66–74, 1997.
- [21] R. R. Selmic and F. L. Lewis, “Deadzone compensation in motion control systems using neural networks”, *IEEE Transactions on Automatic Control*, vol. 45, no. 4, pp. 602–613, 2002.
- [22] R. R. Selmic and F. L. Lewis, “Backlash compensation in nonlinear systems using dynamic inversion by neural networks”, *Asian Journal of Control*, vol. 2, no. 2, pp. 76–87, 2000.
- [23] W. Gao and R. R. Selmic, “Neural network control of a class of nonlinear systems with actuator saturation”, *IEEE transactions on neural networks*, vol. 17, no. 1, pp. 147–156, 2006.

- [24] T. Kohonen, “Self-organized formation of topologically correct feature maps”, en, *Biol. Cybern.*, vol. 43, no. 1, pp. 59–69, 1982, ISSN: 0340-1200, 1432-0770. DOI: 10.1007/BF00337288. Accessed: Nov. 30, 2021. [Online]. Available: <http://link.springer.com/10.1007/BF00337288>.
- [25] B. Kuipers, “The Cognitive Map: Could It Have Been Any Other Way?”, en, in *Spatial Orientation*, H. L. Pick and L. P. Acredolo, Eds., Boston, MA: Springer US, 1983, pp. 345–359, ISBN: 978-1-4615-9327-0 978-1-4615-9325-6. DOI: 10.1007/978-1-4615-9325-6_15. Accessed: Oct. 5, 2024. [Online]. Available: http://link.springer.com/10.1007/978-1-4615-9325-6_15.
- [26] B. Fritzke, “A Growing Neural Gas Network Learns Topologies”, en, 1997.
- [27] M. Toussaint, “A Sensorimotor Map: Modulating Lateral Interactions for Anticipation and Planning”, en, p. 24, 2006, QID: Q48593941.
- [28] R. Peters and O. C. Jenkins, “Uncovering manifold structures in robot’s sensory-data state space”, in *5th IEEE-RAS International Conference on Humanoid Robots, 2005.*, IEEE, 2005, pp. 369–374.
- [29] S. Höfer, M. Hild, and M. Kubisch, “Using slow feature analysis to extract behavioural manifolds related to humanoid robot postures”, in *Tenth International Conference on Epigenetic Robotics*, 2010, pp. 43–50.
- [30] C. Rachuy, “Manifold-based sensorimotor representations for bootstrapping of mobile agents”, Ph.D. dissertation, Universität Bremen, 2020.
- [31] M. Hild, M. Kubisch, and S. Höfer, “Using quadric-representing neurons (qrens) for real-time learning of an implicit body model”, in *Proceedings of the 11th Conference on Mobile Robot and Competitions*, 2011.
- [32] B. Panreck, “Entwicklung einer heuristischen verhaltensregelung für die visuomotorik humanoider roboter”, Ph.D. dissertation, 2022.
- [33] V. Rist and M. Hild, “How to design morphologies. a design process for autonomous robots”, in *International Conference on Simulation of Adaptive Behavior*, Springer, 2022, pp. 3–14.
- [34] F. Benureau, “Self exploration of sensorimotor spaces in robots.”, Ph.D. dissertation, Université de Bordeaux, 2015.

- [35] S. Thrun, “Probabilistic robotics”, *Communications of the ACM*, vol. 45, no. 3, pp. 52–57, 2002.
- [36] J. K. O’regan and A. Noë, “A sensorimotor account of vision and visual consciousness”, *Behavioral and brain sciences*, vol. 24, no. 5, pp. 939–973, 2001.
- [37] R. Saegusa, G. Metta, G. Sandini, and S. Sakka, “Active motor babbling for sensorimotor learning”, in *2008 IEEE International Conference on Robotics and Biomimetics*, IEEE, 2009, pp. 794–799.
- [38] R. S. Sutton, A. G. Barto, et al., *Reinforcement learning: An introduction*. MIT press Cambridge, 1998, vol. 1.
- [39] M. Hild and M. Kubisch, “Self-exploration of autonomous robots using attractor-based behavior control and abc-learning”, in *Eleventh Scandinavian Conference on Artificial Intelligence*, IOS Press, 2011, pp. 153–162.
- [40] M. Hild, “Defying gravity-a minimal cognitive sensorimotor loop which makes robots with arbitrary morphologies stand up”, in *11th International Conference on Accomplishments in Electrical and Mechanical Engineering and Information Technology (DEMI)*, 2013, pp. 23–34.
- [41] M. Hild, B. Panreck, S. Untergasser, and M. Tolksdorf, “(CSL) An adaptive balancing robot with exceptional touch sensitivity driven by an analog sensorimotor loop”, en, *AMAM*, p. 2, Mar. 2021.
- [42] M. Kubisch, C. Benckendorff, and M. Hild, “Balance recovery of a humanoid robot using cognitive sensorimotor loops (csls)”, in *Field Robotics*, World Scientific, 2012, pp. 142–148.
- [43] B. Werner, “Entwicklung eines adaptiven sensomotorischen Algorithmus zur dynamischen Bewegungssteuerung autonomer Roboter”, Diplomarbeit, Humbolt Universität zu Berlin, Humboldt-Universität zu Berlin, Institut für Informatik, 2013.
- [44] S. Bethge, “ABC-Learning: Ein Lernverfahren zur modellfreien Selbst-exploration autonomer Roboter.”, de, p. 83, Jun. 2014.
- [45] M. Janz, “Situationserkennung für autonome Roboter basierend auf Diffusionsprozessen in sensomotorischen Graphen”, Masterarbeit, Freie Universität Berlin, Institut für Informatik, 2015.

- [46] T. S. Parker and L. Chua, *Practical numerical algorithms for chaotic systems*. Springer Science & Business Media, 2012.
- [47] M. Hild, T. Siedel, C. Benckendorff, C. Thiele, and M. Spranger, “Myon, a new humanoid”, *Language grounding in robots*, pp. 25–44, 2012.
- [48] Robotis. “Xl330-m288-t e-manuel”. [Online]. Available: <https://emanual.robotis.com/docs/en/dxl/x/xl330-m288/>.
- [49] STMicroelectronics. “Integrated development environment for stm32”. [Online]. Available: <https://www.st.com/en/development-tools/stm32cubeide.html>.
- [50] L. Kramp, *Development of a Processor-Platform for Humanoid Robots*, Bachelor Thesis, Berliner Hochschule für Technik, 2024.
- [51] V. Rist, *Algorithmische Morphologien für autonome Roboter*, Bachelor Thesis, Berliner Hochschule für Technik, 2022.

Dokumentation der Nutzung von KI-basierten Anwendungen und Werkzeugen

	KI-basiertes Hilfsmittel	Einsatzform	Betroffene Teile der Arbeit	Beschreibung der Eingabe (Prompt)	Bemerkung
1	Deepseek	Formulierungshilfe	Schreiben	Can you formulate this sentence in a more academic, scientific manner: (<i>sentence</i>)	Dies hat geholfen, einige Stellen präziser, professioneller und weniger umfangreich zu formulieren.
2	Deepseek	Codevorlagen für Grafiken	Schreiben	I want (...) kind of layout/graphic. Can you generate a code skeleton in Python, or TikZ for this.	Dies hat das Erstellen von Grafiken mit Python und TikZ für die Arbeit vereinfacht und den Zeitaufwand für die oft lästigen und repetitiven Codesegmente verkürzt.
3	LanguageTool (LT)	Rechtschreibung	Schreiben	(kein Prompt nötig)	Rechtschreibkontrolle mit integrierter KI-Funktion mit Umformulierungsvorschlägen
4	DeepL	Übersetzung	Schreiben	(Deutsches Wort)	Dies hat geholfen korrekte Übersetzung einzelner Wörter, oder Synonyme zu finden.

Nachname : Rist

Matrikelnr. : 6216470

Vorname : Vincent

Hinweise zu den offiziellen Erklärungen

1. Die folgende Seite mit den offiziellen Erklärungen
 - A)** Eigenständigkeitserklärung
 - B)** Erklärung zur Veröffentlichung von Bachelor- oder Masterarbeiten
 - C)** Einverständniserklärung über die Bereitstellung und Nutzung der Bachelorarbeit / Masterarbeit in elektronischer Form zur Überprüfung durch eine Plagiatssoftware

ist entweder direkt in jedes Exemplar der Bachelor- oder Masterarbeit fest mit einzubinden oder unverändert im Wortlaut in jedes Exemplar der Bachelor- oder Masterarbeit zu übernehmen.

Bitte achten Sie darauf, jede Erklärung in allen drei Exemplaren der Arbeit zu unterschreiben.

2. In der digitalen Fassung kann auf die Unterschrift verzichtet werden. Die Angaben und Entscheidungen müssen jedoch enthalten sein.

Zu B)

Die Einwilligung kann jederzeit durch Erklärung gegenüber der Universität Bremen, mit Wirkung für die Zukunft, widerrufen werden.

Zu C)

Das Einverständnis der dauerhaften Speicherung des Textes ist freiwillig.

Die Einwilligung kann jederzeit durch Erklärung gegenüber der Universität Bremen, mit Wirkung für die Zukunft, widerrufen werden.

Weitere Informationen zur Überprüfung von schriftlichen Arbeiten durch die Plagiatsoftware sind im Nutzungs- und Datenschutzkonzept enthalten. Diese finden Sie auf der Internetseite der Universität Bremen.

Nachname: Rist

Matrikelnr.: 6216470

Vorname: Vincent

A) Eigenständigkeitserklärung

Ich versichere, dass ich die vorliegende Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe. Alle Teile meiner Arbeit, die wortwörtlich oder dem Sinn nach anderen Werken entnommen sind, wurden unter Angabe der Quelle kenntlich gemacht. Gleiches gilt auch für Zeichnungen, Skizzen, bildliche Darstellungen sowie für Quellen aus dem Internet, dazu zählen auch KI-basierte Anwendungen oder Werkzeuge. Die Arbeit wurde in gleicher oder ähnlicher Form noch nicht als Prüfungsleistung eingereicht. Die elektronische Fassung der Arbeit stimmt mit der gedruckten Version überein. Mir ist bewusst, dass wahrheitswidrige Angaben als Täuschung behandelt werden.

☒ Ich habe KI-basierte Anwendungen und/oder Werkzeuge genutzt und diese im Anhang "Nutzung KI-basierte Anwendungen" dokumentiert.

B) Erklärung zur Veröffentlichung von Bachelor- oder Masterarbeiten

Die Abschlussarbeit wird zwei Jahre nach Studienabschluss dem Archiv der Universität Bremen zur dauerhaften Archivierung angeboten. Archiviert werden:

- 1) Masterarbeiten mit lokalem oder regionalem Bezug sowie pro Studienfach und Studienjahr 10 % aller Abschlussarbeiten
- 2) Bachelorarbeiten des jeweils ersten und letzten Bachelorabschlusses pro Studienfach und Jahr.

☒ Ich bin damit einverstanden, dass meine Abschlussarbeit im Universitätsarchiv für wissenschaftliche Zwecke von Dritten eingesehen werden darf.

☒ Ich bin damit einverstanden, dass meine Abschlussarbeit nach 30 Jahren (gem. §7 Abs. 2 BremArchivG) im Universitätsarchiv für wissenschaftliche Zwecke von Dritten eingesehen werden darf.

☐ Ich bin nicht damit einverstanden, dass meine Abschlussarbeit im Universitätsarchiv für wissenschaftliche Zwecke von Dritten eingesehen werden darf.

C) Einverständniserklärung zur Überprüfung der elektronischen Fassung der Bachelorarbeit / Masterarbeit durch Plagiatsoftware

Eingereichte Arbeiten können nach § 18 des Allgemeinen Teil der Bachelor- bzw. der Masterprüfungsordnungen der Universität Bremen mit qualifizierter Software auf Plagiatvorwürfe untersucht werden.

Zum Zweck der Überprüfung auf Plagiate erfolgt das Hochladen auf den Server der von der Universität Bremen aktuell genutzten Plagiatsoftware.

☒ Ich bin damit einverstanden, dass die von mir vorgelegte und verfasste Arbeit zum oben genannten Zweck dauerhaft auf dem externen Server der aktuell von der Universität Bremen genutzten Plagiatsoftware, in einer institutionseigenen Bibliothek (Zugriff nur durch die Universität Bremen), gespeichert wird.

☐ Ich bin nicht damit einverstanden, dass die von mir vorgelegte und verfasste Arbeit zum o.g. Zweck dauerhaft auf dem externen Server der aktuell von der Universität Bremen genutzten Plagiatsoftware, in einer institutionseigenen Bibliothek (Zugriff nur durch die Universität Bremen), gespeichert wird.

Mit meiner Unterschrift versichere ich, dass ich die obenstehenden Erklärungen gelesen und verstanden habe und bestätige die Richtigkeit der gemachten Angaben.

28. Juli 2025

