

Modeling and Robust Control of an Eel-Like Robot

A Passivity-based Approach

Dissertation presented by
Martin DOFF-SOTTA

for obtaining the Master's degree in
Electro-mechanical Engineering
Option(s): Mechatronics

Supervisor(s)
Renaud RONSSE, Philippe CHATELAIN

Reader(s)
Caroline BERNIER, Bruno DEHEZ

Academic year 2017-2018

Abstract

This thesis discusses the modeling and robust passivity-based control of an eel-like robot. The eel is modeled as a multi-body system composed of N rigid ellipsoidal segments, evolving in a perfect fluid. The effect of the added inertia is taken into account and an Euler-Lagrange formalism is used to derive the equations of motion. The model is validated against results from the literature and other results from an existing CFD code developed at Université Catholique de Louvain. Then, a robust controller exploiting the passivity properties of the system is derived in the context of input-output stability theory: a Very Strictly Passive controller is synthesized in an optimal way via Riccati equation and Kalman-Yakubovich-Popov lemma (Positive Real lemma), and the robustness of the controller is assessed by the introduction of various disturbances in the system. Finally, a Line-Of-Sight guidance block is added on top of the controller to achieve straight-line tracking, and the simulation results are compared with the literature.

Keywords: Eel-like Robot, Model-Based Control, Dynamical Modeling, Multi-Body Systems, Very Strictly Passive, Robust control, Passivity, Line-Of-Sight Guidance.

Acknowledgments

First of all, I would like to thank my parents for supporting me all over my studies and offering me a radiant future. Nothing would have been possible without their love; many thanks to them for that.

A huge thank you goes to my supervisors Prof. Renaud Ronsse and Philippe Chatelain for their valuable help throughout this work.

Prof. Ronsse adapted the original thesis focus to fit my interest in control theory, offering me to join the "eel-like robot" adventure despite many students were working on the project already. Throughout this year, Prof. Ronsse has been providing me with invaluable advice and has been giving a lot of his time every Wednesday afternoons, stimulating my interest for the subject even more.

Prof. Chatelain also gave me some of his time to discuss the modeling approaches and put me on the right track, giving me the relevant papers.

I also would like to thank Caroline Bernier for her expertise on the simulator and the time she gave me. Thank you also for literally dipping the feet into the icy-cold waters of the pool when it came to experimental work with the eel-like robot ! These experiments would not have been possible without her help.

A special thanks goes to Prof. James Richard Forbes for the fun and inspiring lectures he gave on control theory at McGill University. Prof. Forbes takes teaching seriously and gives a lot of his time to make sure that his students are thrilled enough by the content of the course. Without Prof. Forbes, many of the original contributions of this thesis related to robust control would not have been possible.

I am thankful to Samantha J. C. Flossie for all the discussions we had on control theory and other non-scientific-related topics. I am obviously indebted to her for the support she gave all year round.

Finally, I am grateful to Université Catholique de Louvain for allowing me to study abroad for one semester at McGill University. This thesis would not be what it is without the inspiration I gained from the exceptional people I met there.

Table of Contents

1	Literature Review	4
1.1	Bio-inspired Robotics	4
1.2	Robotic Fish	6
1.3	Eel-like Robots	7
1.4	Modeling and Control of Eel-like Robots	9
1.4.1	Modeling	9
1.4.2	Control	9
1.4.3	Bio-inspired Control: Central Pattern Generators	9
2	Presentation and Experimental Validation of the Simulator VPM-2D	11
2.1	VPM-2D	11
2.2	The Eel-like Robot	13
2.3	Experimental Validation	16
2.3.1	Methodology	16
2.3.2	Results	18
2.3.3	Discussion	23
2.4	Conclusion	25
3	Modeling	27
3.1	Assumptions	27
3.1.1	Assumptions on the fluid	27
3.1.2	Assumptions on the body	28
3.2	Mathematical formalism	29
3.3	Kinematics	31
3.4	Dynamics	32
3.5	Conclusion	38
4	Model Validation	39
4.1	Simulation Results	39
4.2	Energy check	42
4.3	Conclusion	44
5	Passivity-based Robust Control	45
5.1	Mathematical Foundations of I-O Stability Theory, Passivity Theory and Positive Real Matrices	45
5.2	Passivity	50
5.3	Optimal Design of a Robust VSP controller	52
5.3.1	Methodology	52
5.3.2	Trajectory Following	56

5.3.3	Feedback Interconnection and Stability Discussion	57
5.4	Simulations	60
5.4.1	Methodology	60
5.4.2	Results	61
5.4.3	Discussion	66
5.5	SPR control: perspectives and limitations	69
5.6	Conclusion	70
6	Guidance	72
6.1	Line-Of-Sight Guidance	72
6.2	Simulation Results	73
6.3	Conclusion	75
	Appendices	79
A	Programming VPM-2D	80
B	Derivation of the Jacobian, Jacobian Transpose and Jacobian Derivative	82
C	LaSalle's Invariant Set Theorem	84

List of Abbreviations

ARE	Algebraic Riccati Equation
CFD	Computational Fluid Dynamics
CPG	Central Pattern Generator
DBSCAN	Density Based Spatial Clustering of Applications with Noise
DCM	Direction Cosine Matrix
DRL	Deep Reinforcement Learning
EOM	Equations Of Motion
FN	False Negative
FP	False Positive
I-O	Input-Output
KYP	Kalman-Yakubovich-Popov
LOS	Line-Of-Sight
LQR	Linear Quadratic Regulator
LTI	Linear Time Invariant
MBS	Multi Body System
PD	Proportional Derivative
PID	Proportional Integral Derivative
PR	Positive Real
RGB	Red Green Blue
SPD	Spectral Power Density
SPR	Strictly Positive Real
UUV	Unmanned Underwater Vehicle
VPM	Vortex Particle-Mesh
VSP	Very Strictly Passive
ZOH	Zero Order Hold

List of Figures

1	A biological eel, a robotic counterpart (the platform developed at Université Catholique de Louvain), and a simulation snapshot (from the VPM-2D simulator developed at Université Catholique de Louvain). Images courtesy of [1] and [2].	1
1.1	Elmer and Elsie, the "tortoises", had two photo-sensitive cells each, and demonstrated an impressive behavioral repertoire to external stimuli. They are considered as the first ever example of an autonomous electronic robot.	5
1.2	Boxybot, a hybrid crawling and swimming robot from EPFL. Images courtesy of [3].	7
1.3	Envirobot is capable of detecting pollutants present in water in real time. Images courtesy of [4].	8
1.4	Eelume is an example of industrial applications of eel-like robotics: this subsea robot is used in inspections, maintenance and repairs for oil and gas installations. Images courtesy of [5].	9
2.1	N -segment eel-like robot MBS diagram.	13
2.2	Eel-like robot SolidWorks drawing. In the present experimental setup, $N = 4$ links are used with $N - 1 = 3$ actuators operating. Image courtesy of [2]. .	14
2.3	Assembly motor (4) + transmission shaft (2) inserted into the hollow ellipsoidal paddle (12). Image courtesy of [2].	15
	(a) Side view	15
	(b) Top view	15
2.4	Connection between links: the motor shaft is inserted into the hollow transmission shaft of the next link. Image courtesy of [2].	15
2.5	Experimental setup: the $N = 4$ links eel-like robot swimming forward in the LEFT hall pool. Image acquired by a camera (Iphone 6) placed atop the scene.	16
2.6	Comparison between desired and achieved joint space trajectories for joint number 1.	19
2.7	Illustration of a discrepancy occurring in the trajectory tracking for joint 2. .	20
2.8	Successive frames outputted by the color detection algorithm: the green tag attached to the tail of the robot (upper image) is detected and identified by a yellow marker (lower image)	21
2.9	Validation of the color detection experiment: comparison between the expected trajectory of the tail (inferred from joint angles measurement) and the measured trajectory with the color detection algorithm.	22
2.10	Validation of the VPM-2D simulator: comparison between the predicted $x - y$ path of the robot using a VPM-2D simulation and the measured path by color detection of the robot's tail.	22

2.11	Comparison between the predicted static torque (VPM-2D) and the measured static torque at the last joint.	23
3.1	N -segment eel-like robot kinematic.	31
4.1	Prescribed kinematic pattern at the joints (controllable or shape variables)	40
4.2	Trajectory in the body variable space.	41
4.3	Nondimensionalized trajectory of the central body centroid. In blue and red: viscous-based models ; in green and black: inviscid Kirchhoff theory model.	41
4.4	Conservation of the total mechanical energy in the absence of non-conservative forces.	43
4.5	Variation of the total mechanical energy under actuator torques.	43
5.1	Feedback interconnection of a passive system with a VSP system. Passivity theorem states that the closed loop is input-output stable. The signals $\mathbf{d}$ and $\mathbf{n}$ represent external inputs (e.g disturbances and noise).	48
5.2	Feedback interconnection of a passive system with a linear SPR controller. Passivity theorem states that the closed loop is input-output stable.	58
5.3	Comparison of PD and SPR controller in the event of the addition of disturbances to the system. Both controllers were initially fine tuned in the absence of external inputs.	61
5.4	Evolution of the mean error between desired and achieved trajectory for different disturbance SPD.	62
5.5	Comparison of PD and SPR controller in the event of the addition of noise to the system. Both controllers were initially fine tuned in the absence of external inputs.	62
5.6	Effect of initial disturbances on the closed loop.	63
5.7	Effect of torque saturation on the closed loop.	64
5.8	Effect of unmodelled dynamics (VPM-2D simulation) and initial disturbances on the closed loop.	64
5.9	Zoom on Figure 5.8.	65
5.10	Effect of unmodelled dynamics (VPM-2D simulation) and actuator saturation on the closed loop.	65
5.11	Effect of saturation in the x, y -path.	66
5.12	Effect of varying the penalty matrix on the performances of the closed loop.	67
5.13	Effect of varying the penalty matrix on actuator torques.	67
6.1	Principle of LOS guidance: the heading of the system is steered toward the look-ahead target.	73
6.2	Closed loop: outer loop responsible for path following (issuing a desired trajectory in joint space); inner loop for achieving the desired trajectory. .	73
6.3	Straight-line following, starting with a cross-track error of $p_y = 3$ m and with an initial heading direction aligned with the path.	74
C.1	Lyapunov stability.	84

List of Tables

6.1	Guidance parameters used in the simulation	74
6.2	Geometric parameters of the eel-like swimmer	74

Introduction

BIO-INSPIRED robotics gives rise to a prolific literature concerned with the modeling, implementation and control of various types of systems taking advantage of their biological counterparts [6]. In particular, recent advances have been achieved in the field of underwater robotics with autonomous systems capable of mimicking the locomotion of biological fish [7], [8]. These robots inspired from nature are now able to move more efficiently in aquatic environments, by contrast to traditional systems relying on energy-consuming man-made propellers [9], [8]. The applications are numerous: marine transportation, underwater installation maintenance, scientific exploration, save and rescue missions or underwater warfare with increased maneuverability.

Not only bio-inspired underwater robots open the way to tremendous increases of efficiency in marine locomotion, but their study can also improve our understanding of biological systems in many ways. Indeed, it is widely known that there exists a synergy between robotics and biology – the modeling and design of fish robots offering a better understanding of the swimming mechanism among biological fish, such as lamprey or eel [10], [11], [12].

Especially, this thesis covers the modeling and control of an eel-like robot (see Figure 1) for its potential applications in underwater locomotion and better understanding of biological systems.

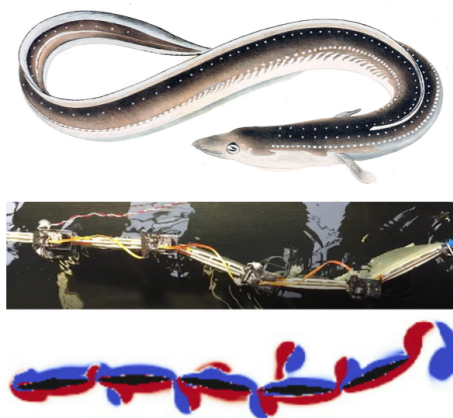


Figure 1: A biological eel, a robotic counterpart (the platform developed at Université Catholique de Louvain), and a simulation snapshot (from the VPM-2D simulator developed at Université Catholique de Louvain). Images courtesy of [1] and [2].

The focus of this work is on control design, more specifically the design of a model-based controller in order to achieve efficient locomotion in the presence of disturbances and unmodelled dynamics. Indeed, a robot swimming underwater is submitted to various external disturbances (waves, eddies, streams, ...) that may divert it from its goal. Moreover, due to the complexity of the fluid-structure interaction, developing a workable¹ and realistic model of the hydrodynamic forces acting on the robot is often beyond reach. For that particular reason, it is highly desirable to design controllers that are robust to model uncertainties (unmodelled dynamics, parameters uncertainty), so that model imperfections are alleviated by flexible control architectures.

Consequently, this thesis proposes a passivity-based² controller that can achieve trajectory tracking in joint space (inner-loop control) while guaranteeing a certain degree of robustness to unmodelled dynamics and external inputs. Two main tools will be used throughout this work in order to carry out this task: a simplified inviscid simulator built on MATLAB / SIMULINK aiming at designing the controller at no cost (quick simulation time) and a high-fidelity viscous-flow simulator developed at Université Catholique de Louvain (VPM-2D), used to validate the design. A consistent part of this work is thus dedicated to respectively develop and validate these simulators, before tackling the control design problem. Finally, the developed controller is integrated in a complete guidance loop in order to achieve path following.

The objectives of this thesis are summarized as follows:

- develop a simplified but realistic model of an eel-like swimmer. By "realistic" we mean that the model can qualitatively reproduce the behavior of more complex models, based on Computational Fluid Dynamics (CFD) methods (e.g the VPM-2D simulator). By "simple" we mean that the hydrodynamic efforts can be formulated analytically and the model used in a control-oriented perspective.
- validate the model against literature results (from which the model is inspired) and draw conclusions on the comparison with high-fidelity simulators such as VPM-2D.
- using the model presented, design a robust inner-loop controller and test it on both simulation environments (inviscid simulator and VPM-2D).
- close the loop with a guidance algorithm.

In that respect, this thesis starts by a review of the state of the art in fish-like and anguilliform robotics. Additional topics are also covered such as the applications of such robots, the control and modeling aspects, and a quick review of bio-inspired robotics.

Then, the resources available for this project are introduced in Chapter 2: VPM-2D is presented, along with the robotic platform developed in the context of a former master's thesis. Experimental tests on the robot are then carried out: predictions from the CFD simulator are compared against experimental results, and a localization method is also implemented.

¹By workable, it is intended that the model is to be used in real-time or can be easily used in a control design perspective. Analytic models are thus "workable" design solutions, by contrast to computationally intensive numerical methods.

²Passivity is a control-theoretic concept characterizing systems whose energy can be bounded by some known limits; to that extent, passivity is intrinsically related to stability of the system of interest and a variety of robust control approaches exploit this model-based property (more on this later).

Chapter 3 is dedicated to the modeling of a N -link eel-like robot using the perfect fluid assumption. The kinematics and dynamics of swim are established. Validation of the model by simulations is then investigated in Chapter 4, including energy checks and comparison of the presented model with results from the literature.

The core of this thesis is covered in Chapter 5, where the design and implementation of an inner-loop controller is discussed. The passivity properties of the system are first unveiled from the model, before being exploited for the synthesis of a robust controller. The obtained controller presents a hybrid form, combining the characteristics of an optimal state feedback controller (in a Linear-Quadratic-Regulator fashion) with the robustness properties guaranteed by the so-called passivity theorem. The controller is simulated and its performances are assessed by comparison with a benchmark controller in the presence of disturbances.

Finally, Chapter 6 covers the implementation of a guidance controller, and the results are compared with the literature.

Chapter 1

Literature Review

The interest for fish-like robots arises from the amazing performances of their biological counterparts. They can achieve "acceleration up to 20 times the gravity, their speed exceeding 70 km/h" [13]. Their remarkable maneuverability – i.e., 180° turn without slowing down and radii of curvature about 10 times their length – far exceeds the capabilities of man-made vehicles. Indeed, the efficiency of the swimming mechanism is about 10 times higher than blade-propelled submarines [13].

Fish are commonly classified by the way they swim; many sorts of swimming behaviors have been reported: carangiform, sub-carangiform, ostraciform, anguilliform, and so on [14]. The latter characterizes the swim of lampreys or eels, and is of particular interest due to the high energy efficiency achieved by these fish [15]. Especially, the focus of this thesis is on anguilliform swimming and the control of eel-like robots, for their advantageous characteristics acquired from millions of years of evolution. An eel-like swimmer achieves propulsion by propagating oscillatory waves of increasing amplitude along its body, from head to tail. Maneuverability benefits from the large deformations withstood by their body, along with the high inter-segmental coordination of their spinal chord. Studies have shown that eels are 5–6 times more efficient than any other fish – one of the main mystery in animal kingdom being their long annual migration from coastal Europe to Sargasso sea (*circa* 6000 km, [15]).

These incredible performances motivate the current research on biomimicry and the understanding on how nature can be mimicked in order to reproduce evolutionary-selected patterns in our robotic systems. Therefore, our investigation starts by a review on bio-inspired robotics in general, before covering fish robots and eel-like robots in particular. Finally, this Chapter reports advances in eel-like locomotion in a control-oriented perspective with the investigation of key concepts such as Central Pattern Generators (CPG) or dynamic modeling.

1.1 Bio-inspired Robotics

The study of biological organisms and self-organization can strongly benefit the design of autonomous robots. While industrial robots operate in highly structured and predictable environments with a focus on speed, precision and cost-effectiveness, bio-inspired robots have to cope with wild, changing, uncertain situations and react quickly in unstructured environments [6].

The complexity of the biological world is reflected in the many types of robots developed

so far: these range from molecular and cellular reproduction to primates, including aquatic animals and the insect world [6].

The worked-out principles of biological systems are transferred to robot design by mimicking the key characteristics of biological structures [16]. In 1949, pioneer of cybernetics Grey Walter explicitly used behavior-based concept for his two electromechanical Tortoises, see pictures in Figure 1.1, [17].

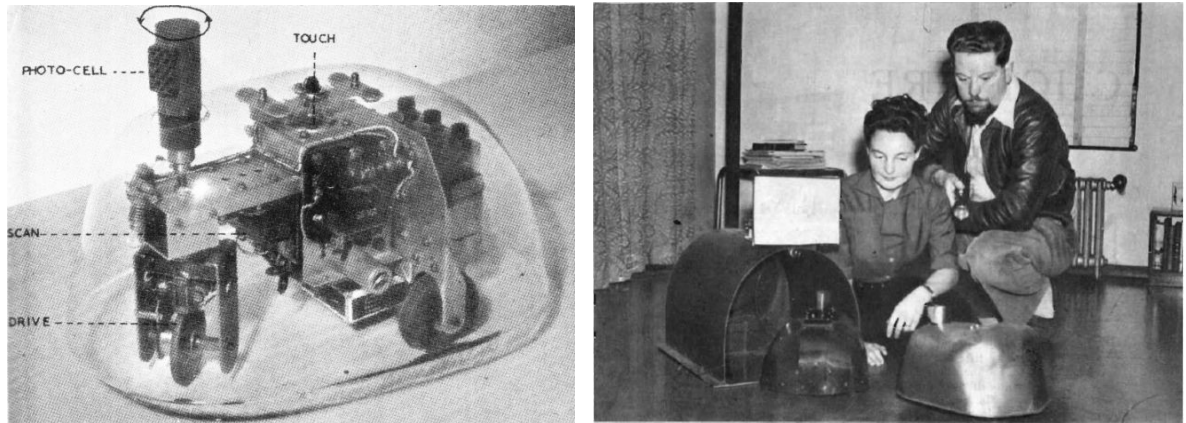


Figure 1.1: Elmer and Elsie, the "tortoises", had two photo-sensitive cells each, and demonstrated an impressive behavioral repertoire to external stimuli. They are considered as the first ever example of an autonomous electronic robot.

The field has grown quickly over the years. Researches focusing on locomotion have been carried out e.g. on the switching between swimming to walking with an amphibious spinal-cord-model robot [18], or on undulatory vehicles with a neural circuit-based controller (or CPG-like model) driving muscle-based linear actuators through a basic biomimetic plant [19], [20].

Neural modeling triggers research e.g on spatial memory formation and navigational skills among rodents [21], thus showing an example of how biology can benefit from robotics.

In [22], Pfeifer reports a growing interest in the notion of "embodied agent" dealing with highly connected sensors, actuators, limbs and neural system. The concept of morphological computation is coined to define the correlation between brain, body and environment [23]. Embodied agents are simpler, more robust and adaptive than those based on classical control schemes and are capable of cooperation in order to accomplish specific tasks with much less computational power [24].

In [25], morphologically and behaviorally different robots are used to illustrate the influence of morphology and sensor organisation on locomotor capabilities of an autonomous agent.

Orientation and navigation behaviors for homing are studied exploring optical flow techniques [26] or using morphological robots able to track the sound of insects like crickets [27].

Adaptive machines capable of changing their shape in order to achieve their tasks are investigated in [28]. They are built with many degrees of freedom and different materials, taking nature as a "design teacher".

1.2 Robotic Fish

A progress report of fish-inspired mechanical systems is reviewed in this section.

Such autonomous vehicles are essential tools for example in environmental monitoring systems [4] or to estimate the level of pollution in water [29]. Biomimetic robotic fish are also useful as an educational and research tool for innovative approaches in temperature sensing, control, navigation and communication [30]. Robotic fish may also find applications in marine or military fields where hazardous waters, mine countermeasures, toxic agents on the cluttered sea bed, leakage in oil pipelines, etc., require fast response miniaturized solutions. [31]

Efforts to build robots that mimic the kinematics of undulatory fins are essential to improve the maneuverability of Unmanned Underwater Vehicles (UUV) with highly responsive multidirectional movements. Improved agility and propulsion are the biggest advantages of robotic fish over their propeller-driven cousins.

Robotic fish intends to use characteristics of the real fish, but not all fish fall in one determined group. There is a variety of biological mechanisms for propulsion in water (fish locomotion). The major form of locomotion among fish is anguilliform swimming (e.g eels and morays) achieved by means of sinusoidal lateral movements of the body and median fins. In sub-carangiform swimming (e.g salmonids) the propulsion is provided by the tail and the posterior half of the body. Then comes carangiform swimming (e.g makrell) in which propulsion is achieved mostly by means of rapid oscillations of the tail. By contrast, thunniform swimming (e.g tunas) allows a rapid advance thanks to mid-body muscles. Finally, ostraciiform swimming (e.g boxfishes) is a type of locomotion performed by inflexible fish and characterized by undulation restricted to the caudal fin [14].

RoboTuna, the first fish robot ever, was built in 1994 at MIT. To mimic the undulatory swimming, the eight-link robot is driven by means of external brushless motors. A flapping foil exploiting external fluid forces is used to produce higher accelerations. The aim of the work is to study fluid-mechanics principles, propulsion systems and the role of vortices in efficient swimming [32]. Improvements in terms of obstacle avoidance, up-down motion and vorticity control propulsion have since then been developed by the Draper Lab (MIT) exploiting the body and tail kinematics of RoboTuna.

The VCUUV (Vorticity Control UUV) – named after the vorticity control mechanism used by natural fish to propel and maneuver – is a replica of the yellowfin tuna, following its morphology and kinematics. Four active links are used to perform the desired body undulations, allowing significant improvements of the maneuvering capabilities by contrast to RoboTuna [9].

A carangiform fish is mimicked with a three-link robot by a research team from California Institute of Technology in 2000, see [33], [34]. The robot consists of a rigid body part at the front, a large wing-like tail at the rear, and a slender stem (or caudal peduncle) which connects the two. Experimental tests are then conducted to test the quasi-steady flow model for predicting the thrust generated by the flapping tail. The fish-like swimmer is also able to "perform a very rapid turn in a small space with a single dramatic body motion" [33], [34].

Boxybot (see picture in Figure 1.2) is a fully autonomous robot developed at Ecole Polytechnique Fédérale de Lausanne [3]. The control architecture is based on a Central

Pattern Generator (CPG), implemented as a network of coupled oscillators able to generate coordinated patterns of rhythmic activity with simple control inputs¹. It is a swimming and crawling robotic fish using ostraciiform (and labriform) modes. It is equipped with caudal and pectoral fins used for the ostraciiform mode, while the caudal fin can be used as a rudder in labriform mode. The robot is equipped with light sensors, allowing to follow a bright light moving slowly. The CPG implementation allows to switch easily from forward to backward gait, or from swimming to crawling.

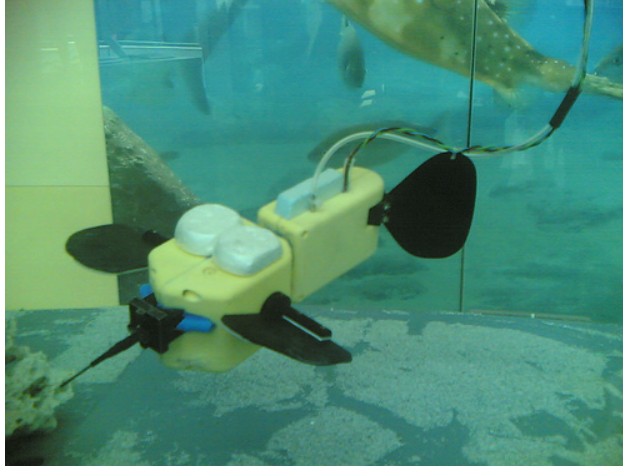


Figure 1.2: Boxybot, a hybrid crawling and swimming robot from EPFL. Images courtesy of [3].

A self-contained soft-bodied robotic fish capable of rapid continuum-body motion and performing sudden escape maneuvers is described in [36]. The compliant body emulates the slender anatomical form of a fish. Pressurized fluids are used to power fluidic elastomer actuators. The stored fluids are distributed to the robot's actuators by both a gas regulation mechanism and delivery system. Its closest precursor, the remote-controlled Airacuda fish from FESTO, can also perform static diving and swimming thanks to onboard pneumatic actuation system. The four pneumatic muscles allow to achieve a flexible body movement and steering [37].

1.3 Eel-like Robots

Anguilliform (eel-like) fish achieve locomotion by propagating undulations of increasing amplitude along their body from head to tail.

This locomotion gait transposed to a robotic platform is a highly desirable solution for achieving flexible and quick maneuvers, with applications to deep-sea exploration or maintenance of highly constrained and inaccessible terrains [38].

This section reviews some of the eel-like robotic platform investigated in the literature.

Amphibot II (a new version of a two-dimensional-motion-capability snake-like robot) is capable of crawling and swimming. It is a seven actuated segments robot with an on-board CPG based on a system of amplitude-controlled phase oscillators generating a travelling

¹Various designs for such neural control architectures (capable of producing the patterns of stable and permanent oscillations necessary for eel-like swimming) were developed in [35].

wave propagated from head to the tail [39].

In [40], an analytic model of an eel-like swimmer based on the Large Amplitude Elongated Body Theory is combined with a Newton-Euler-Featherstone approach for integrating the equations of motion. A robotic platform, AmphiBot III, is developed to test the hypotheses of the simulation environment. The actuated segments are mechanically and electrically connected by an elastic connector. The head segment contains the locomotion control board and an embedded radio device. The seven modular body segments contain each a power board implementing a Proportional Derivative (PD) position controller, a Direct Current gear-motor driving the elastic connector and a battery. The tail segment supports a rigid caudal fin.

ACM-R5 is a three-dimensional amphibious snake-like robot developed for search and rescue missions. It is an extension of the 3-D active cord mechanism HELIX (full-body helical motion) in [41]. ACM-R5 is composed of several modules with yaw and pitch degrees of freedom thanks to two pairs of sector gears located perpendicularly on its both sides [42].

The 3D-printed « Envirobot » (see picture in Figure 1.3), from Ecole Polytechnique Fédérale de Lausanne, collects environmental informations to detect sources of water pollution. It consists of a bio-inspired lamprey-like robot with a head-unit, containing most of the processing capabilities. The numerous modules of the robot contain each a small electric motor but also physical, chemical and biological sensors. Some modules have tiny chambers that host bacteria or fish cells (bactosensor chip) whose reactions with water pollutants can be analyzed in real time [4]. As a major enhancement, visual capabilities have been added to Envirobot using a head module with two wide-angle onboard cameras [43].

In, [44], the capabilities of Envirobot in terms of thrust, power consumption and velocity are compared with blade-propelled UUV.



Figure 1.3: Envirobot is capable of detecting pollutants present in water in real time. Images courtesy of [4].

Eelume is a subsea eel-like autonomous agent capable of performing operations of inspections, maintenance and repairs for oil and gas installations at reduced costs. Able to change shape and produce lateral and longitudinal thrust forces, the robot is tether-free and camera-equipped, see Figure 1.4 [5].



Figure 1.4: Eelume is an example of industrial applications of eel-like robotics: this subsea robot is used in inspections, maintenance and repairs for oil and gas installations. Images courtesy of [5].

1.4 Modeling and Control of Eel-like Robots

1.4.1 Modeling

Modeling of eel-like robot is abundantly referenced in the literature. For example, in [40], Lighthill theory (1960) for a slender body moving in a fluid is used to establish the Equations Of Motion (EOM) with a Newton-Euler formalism. A different approach based on the resolution of Laplace equation for a potential and inviscid flow is used in [45], allowing to derive the Lagrangian of the system fluid and body as depending only on the body configurations. The action of the fluid is modeled naturally as an addition to the inertia of the body.

More sophisticated 3D models also exist in the literature along with models considering the system as a continuum, [46]. By contrast, a very simple control-oriented model is proposed in [47].

1.4.2 Control

Regarding the control problem, most of the results are inspired from (ground) snake robotics research from the 1990's, see [48] for a review. Most of the approaches reviewed are based on simple control strategies and model-free control, due to the difficulty to capture the complexity of the model. For example, in [38], a PD controller combined with Line Of Sight (LOS) guidance is used to achieve trajectory following. In [49], an approach based on integral LOS is used, allowing to follow trajectories in the presence of disturbances (streams) thanks to integral action. This approach is further extended in [50] using an inner-loop model-based controller: starting from the simplified control-oriented model developed in [47], the nonlinear dynamics is cancelled through feedback linearization in order to achieve joint space trajectory tracking.

1.4.3 Bio-inspired Control: Central Pattern Generators

Animal locomotion is fascinating: neurobiological researches has found that the rhythmic behavior induced by locomotion is not regulated by a sensory consequence of moving body parts, but by neural networks properties [51]. These neural networks responsible

for locomotion among vertebrates are called Central Pattern Generator (CPG). They are nonlinear coupled oscillators capable of generating and modulating stable oscillatory patterns involved in locomotion activities. Implementation of a CPG for controlling motion thus yields a robust bio-inspired solution for generating stable oscillatory signals [52]. These oscillatory signals can then be used as an input for an inner-loop controller in order to achieve rhythmic activities such as swimming or crawling.

The advantage of CPG over traditional methods is the low dimensionality of the input used for the modulation: while traditional approaches use sensory feedback information such as speed, position, gait pattern, etc., CPG inputs are usually very simple. However, most of the time, CPG have to be tailor made for specific applications [3].

The CPG, like its biological counterpart, can produce modulated and coordinated patterns of rhythmic activity as swimming and crawling such in [3], the agility of the BoxyBot robot is demonstrated using a CPG-based control approach. The control architecture is shown to achieve stable rhythmic patterns and rapidity switch between different locomotion behaviors with smooth trajectories despite abrupt changes of control parameters. Multiple others robots have been designed using a CPG-based architecture. For example, a CPG-based controller is implemented for the locomotion control of a quadruped robot in [53]. The lying down behavior observed among cats and dogs could be reproduced using this approach.

In [54], flexible and precise turning maneuvers in both forward and backward swimming are implemented for a sub-carangiform robot with CPG control.

An other application of CPG for bipedal locomotion control is achieved with a 23 degrees of freedom simulated humanoid robot. The novel CPG system can learn walking trajectory signals in a supervised learning framework: « Using adaptive rules implemented as differential equations, parameters such as intrinsic frequencies, amplitudes, and coupling weights are automatically adjusted to replicate a teaching signal." [55] When the teaching signal is removed, the CPG is capable of reproducing the learned pattern and achieve walking gait.

Chapter 2

Presentation and Experimental Validation of the Simulator VPM-2D

THE simulator VPM-2D developed at Université Catholique de Louvain is a high-fidelity CFD code combining a Vortex Particle-Mesh method with a Multi-Body System solver for articulated structure-fluid interaction. This Chapter aims at validating the simulator via experimental results. After introducing the simulator and the robot serving as a test-bench, the experimental methodology is described. Then, the results of the experiments are presented and discussed, allowing to make some suggestions on future design and give a direction to this work.

This Chapter has a special place in this thesis as it deals almost exclusively with experimental work. In that respect, a particular emphasis is put on the separation between the experimental methodology, the presentation of results, and the discussion.

2.1 VPM-2D

The VPM-2D simulator ("Vortex Particle-Mesh for 2D problems") is a CFD code developed at Université Catholique de Louvain aiming at solving complex flows arising from the interaction of fluid-solid structures. Originally developed by Grazzola in [1] for studying the interaction between a fluid and a body with prescribed deformations, the simulator has since then been extended to tackle *a priori* unknown deformations with the treatment of systems of articulated bodies by Bernier in [56].

In its current form, VPM-2D is thus a hybrid solver for 2D problems that combines a Vortex Particle-Mesh (VPM) method with a Multi-Body System (MBS) solver.

VPM methods for solving incompressible flows past deforming bodies consist in a Lagrangian discretization of the Navier-Stokes equations in their vorticity formulation¹ [56]. The classical VPM approach considers the kinematics of the moving bodies as known *a priori*, whereas the hybrid approach used here first computes the hydrodynamic efforts acting on the structure with the VPM method and then solves the internal dynamics of the articulated bodies with a MBS solver.

¹The velocity field can then be recovered from Poisson equation

The level of refinement intended here aims at taking into account highly nonlinear phenomena occurring in the flow, such as turbulences, wake of other moving structures and vortex shedding due to obstacles.

The hybridization of the VPM and the MBS solvers yields the following algorithm [56]:

- *Velocity update.* At first, the velocity field is updated by solving the Poisson equation relating velocity and vorticity fields.
- *Projection.* Then, in the so-called projection step, the flow is solved by considering that the fluid is unconstrained by the immersed bodies. In other words, the fluid is set free to move in the region occupied by the solid, as if it was vacant. This results in an addition to the momentum and angular momentum of the bodies.
- *Hydrodynamic and hydrostatic torques.* The hydrodynamic torques acting at the joints of the bodies are computed from the results of the projection method at the current time step and the penalization from previous time step. In the case of neutrally buoyant systems, the hydrostatic forces and momenta are zero, otherwise they have to be taken into account at this stage.
- *MBS solver.* Based on the actuation torques (generated by a controller) and the hydrodynamic torques obtained previously, the dynamic equations of the articulated bodies are integrated with an adaptive-time-step fourth-order Runge-Kutta-Fehlberg method. The new configuration of the system is thus obtained.
- *Shape update.* The motion of the bodies during the current time step is translated into a characteristic function describing the new shape of the articulated system.
- *Brinkman penalization.* The no-slip condition at the interface fluid-solid is enforced using a Brinkman penalization of the velocity field. Basically, this methods consists in adding a penalization term in the Navier-Stokes equations to bring the velocity of the fluid inside the region occupied by the solid to a prescribed body velocity. A penalization of the vorticity is also performed; the penalized forces computed will be used in the next time step for the computation of the hydrodynamic torques acting on the bodies.
- *Vorticity update.* Finally the vorticity field is updated by evaluating the spatial derivatives with second order centered finite difference schemes.

In the context of this thesis, the algorithm above is applied to eel-like swimmers, see Figure 2.1 for a typical MBS diagram. The articulated structure consists of N ellipsoidal rigid bodies assembled in an open chain with rotational joints. The flow is solved in the 2D plane containing the ellipses. The torques τ_i applied at the joints can be specified at each time step (using a controller). The following geometric parameters are reported on this Figure: the link length $2a$, the minor semi-axis b , the tip-to-joint distance c , the number of links N and joints $N - 1$. The latter defines the number of generalized coordinates $n = N + 2$:

- three generalized coordinates are used to specify the pose of the first link: x, y and θ .
- $N - 1$ generalized coordinate are used to describe the (relative) rotation of the $N - 1$ joints: $\phi_1, \phi_2, \dots, \phi_{N-1}$

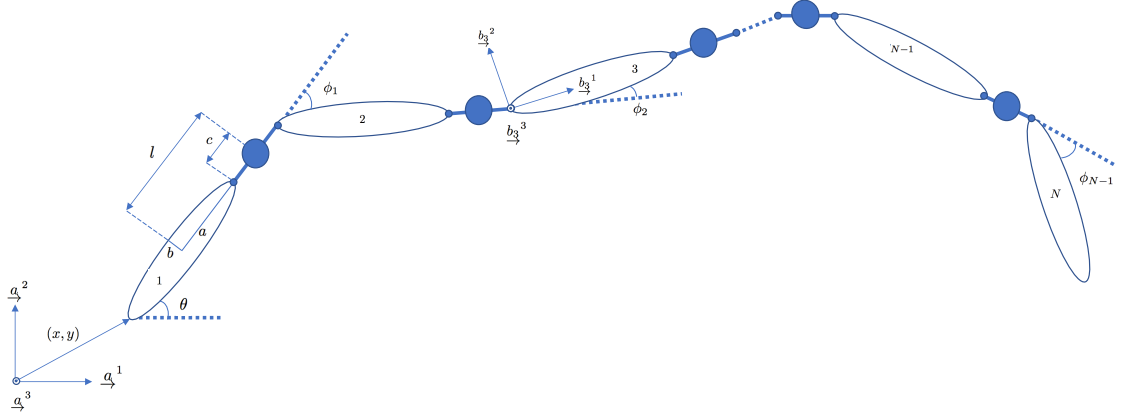


Figure 2.1: N -segment eel-like robot MBS diagram.

Also the following physical parameters are of paramount importance to characterize the flow: the density of the fluid ρ and the kinematic viscosity of water ν . The density of the solids ρ_s should also be specified.

The next section aims at presenting a robotic platform to reproduce in an experiment the conditions of the VPM-2D simulator.

2.2 The Eel-like Robot

In order to validate the simulator, an eel-like robot was designed by former master's students [2]. The simplified design matches the conditions of the simulator and the ease of assembly of the different components makes it an ideal platform to critically assess the simulations against experimental tests.

From the functional analysis performed by the students, the robot supposedly has the following characteristics/abilities in order to comply with the assumptions of the simulator²:

- critical parts (actuators, wire connectors) are made waterproof and emerged.
- the robot is stable in water and neutrally buoyant.
- the mechanical parts are easy to assemble/disassemble.
- the robot is free to move forward in the pool of the LEFT Hall at Université Catholique de Louvain.
- the bodies have ellipsoidal geometries.

²The experimental tests in the following sections will critically assess if the robotic platform indeed complies with these assumptions.

- a significant height is given to the ellipsoidal bodies evolving in shallow water in order to avoid simulation-unmodelled boundary effects³.
- the robot is equipped with a single-board computer capable of controlling the actuators and acquiring data.

The robot that has been designed following the guidelines above is presented in Figure 2.2. The robot consists of N links free to rotate about the vertical axis. In the present experiment, $N = 4$ links only are used. Each link consists of the assembly of a transmission shaft and a Dynamixel AX-12A motor (maximum torque: 1.6 Nm). Then, the assembly is inserted into a hollow ellipsoidal paddle as shown in Figure 2.3. The links are jointed by connecting the motor shafts to the hollow metal rods of the next link as seen in Figure 2.4.

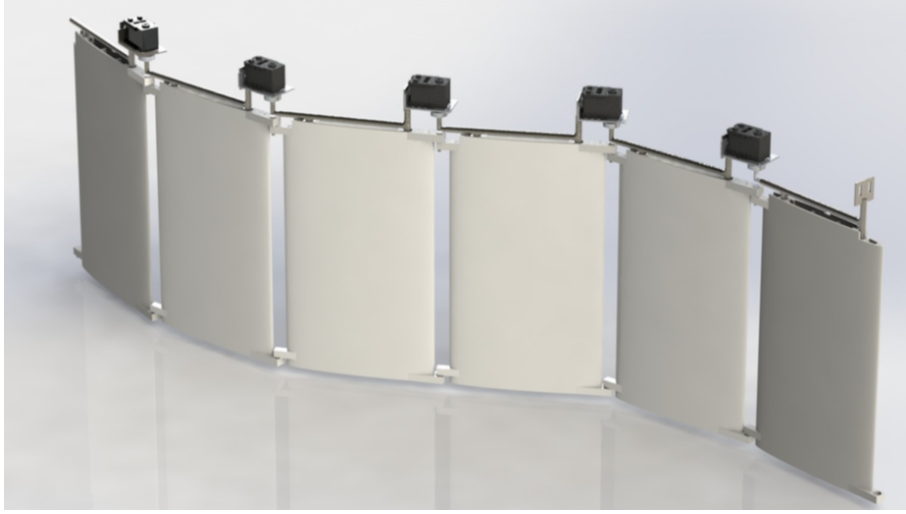


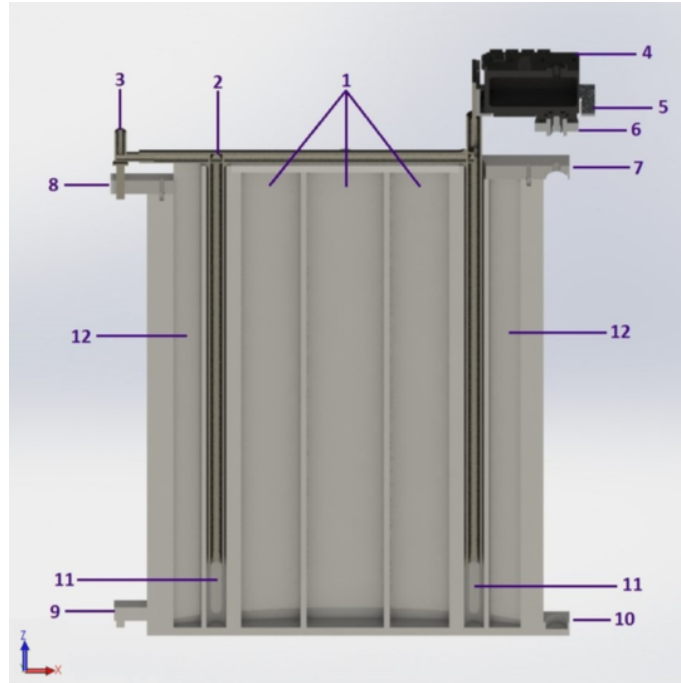
Figure 2.2: Eel-like robot SolidWorks drawing. In the present experimental setup, $N = 4$ links are used with $N - 1 = 3$ actuators operating. Image courtesy of [2].

The main dimensions of the robot are (referring to Figure 2.3): major semi-axis of an ellipse $a = 0.1$ m, minor semi-axis $b = 0.01$ m, tip-to-joint distance $c = 0.01$ m, height $h = 0.3$ m. The detailed drawings of the robot with all dimensions may be found in [2].

The single-board computer used for controlling the Dynamixel motors and acquiring the data is the Arduino Micro. It is powered by an ATX power supply unit connected to the grid. In practice all computations are performed offline for the ease of data acquisition. The Arduino board and the Dynamixel motors are thus interfaced with a long cable.

Finally each motor is assigned with a color tag and the robotic platform is put in the LEFT hall pool, as seen in Figure 2.5. Each operating motor is assigned an identification number from 1 to 3 (from head to tail). In order to localize the robot with respect to a reference frame, the x, y -frame is collocated with the top of the image as shown.

³Indeed, the VPM-2D being a 2D simulator, it is desired to reproduce 2D-like conditions in the physical world.



(a) Side view



(b) Top view

Figure 2.3: Assembly motor (4) + transmission shaft (2) inserted into the hollow ellipsoidal paddle (12). Image courtesy of [2].

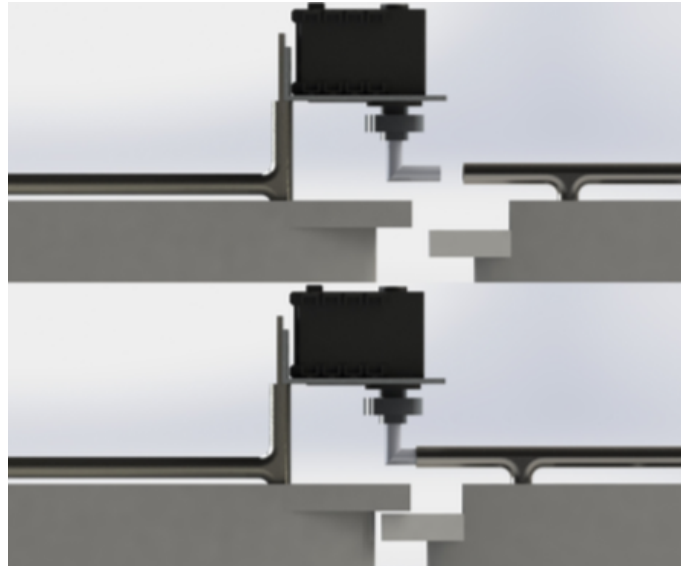


Figure 2.4: Connection between links: the motor shaft is inserted into the hollow transmission shaft of the next link. Image courtesy of [2].



Figure 2.5: Experimental setup: the $N = 4$ links eel-like robot swimming forward in the LEFT hall pool. Image acquired by a camera (Iphone 6) placed atop the scene.

2.3 Experimental Validation

2.3.1 Methodology

2.3.1.1 Objective

The objective of the experiment is twofold: on the one hand it aims at validating the VPM-2D simulator, on the other hand it aims at assessing the suitability of the platform designed. The following key-points are achieved:

- buoy (keep afloat) the robot in shallow water by proper ballasting.
- achieve inner-loop control, i.e joint angles control of the actuators.
- acquire torque and joint angle measurements.
- achieve localization of the robot.

Collecting and processing the various data allows to compare the behavior of the robot in the pool with the prediction from the simulator – namely the predicted joint torque and predicted position of the robot are compared with the experimentally-acquired data. In addition, the platform is evaluated throughout the experimental process based on qualitative criteria (such as "how easy can the robot be stabilized in water") and quantitative criteria (e.g the matching between the desired joint angle trajectory and the one achieved by the Dynamixel via the built-in controller).

2.3.1.2 Preliminaries

The robot designed by Barbason and Vanderweggen (see [2]) is used without any additional structural modifications. The Arduino code is adapted slightly to modify some parameters and acquire desired data.

At first, the wire connections and communication between the micro-computer and the actuators are tested outside water. Then, the robot is assembled by mechanically

connecting all 4 segments together and inserting the transmission shafts supporting the motors into their paddle casings. The critical parts (wire connectors and actuators) are sealed with a combination of plastic foils and tape. Finally, the robot was put in the water and kept afloat by properly ballasting the hollow paddles.

2.3.1.3 First Experiment: Inner-Loop Control

The first experiment aims at assessing the ability of the Dynamixel to follow a trajectory in joint space thanks to its built-in controller. Unfortunately, it is not possible to modify the built-in controller without rewriting the firmware (which is hard to do and may damage the Dynamixel). The user can only specify a desired position (position control), with no possibility to do torque control or implement a custom controller. The following trajectory in joint space is chosen to achieve forward anguilliform motion⁴:

$$\phi_i^*(t) = \alpha(t)g(i, N) \sin(\omega t + i\delta), \quad (2.1)$$

where ϕ_i^* is the desired relative joint angle of actuator i , $g(i, N) = \frac{i}{N}$ is a gait pattern for anguilliform motion with an incremental increase in amplitude from head to tail, $\omega = 2\pi f$, where $f = 0.25$ Hz is the frequency and $\delta = -\frac{\pi}{2}$ the phase shift between the joints. For practical reasons, we use an amplitude of the form: $\alpha(t) = 1.49(1 - \exp(-t/3))$ rad for a gradual increase of the oscillations with time. By doing so, the disturbances generated in water at start-up are limited. Note that the law given in equation (2.1) still has to be adapted to fit the format of the Dynamixel registers (see the datasheet [57]).

When the robot is kept afloat in steady water (with the presence of the user in the pool⁵), the micro-controller is turned on and starts executing the code. ; the joint angles are measured in real time by the Dynamixel built-in sensors.

2.3.1.4 Second Experiment: Color Detection for Localization

In this second experiment, video processing with feature detection is used to localize the robot thanks to a video camera placed atop the scene. The algorithm chosen for the processing is color detection for its relative simplicity: a green tag is placed on the tail of the robot and the images are filtered out to obtain only green pixels. Because color detection is highly sensitive to noise and lighting conditions, a robust and fast clustering algorithm named DBSCAN (Density-Based Spatial Clustering of Applications with Noise, [58]) is applied on top of that to eliminate potential outliers and noise. Finally, in order to extract distances information in an image, a known length object is placed in the scene and all measured distances in pixel space are converted into physical space by comparison.

The color chosen for the tag is green as it is unlikely that a green object present in the pool interferes with the detection process. The color tag is attached to the tail of the robot but any other segment position can be inferred from there with joint angles information. The known-length object used is a tape of 20 cm placed on the side window of the pool.

The joint angles of the robots are controlled (inner-loop control) with the same desired trajectories as previously.

⁴see Chapter 6 for a presentation of this guidance law.

⁵The user has to continuously follow the robot to prevent it from sinking in water, with a risk for the actuators.

Note that the localization was not performed in real time but only in post-processing for validation purpose. Further work would involve running the color detection algorithm in real time – probably with an external computer linked to the camera. The camera can be attached to a robotic movable frame capable of following the robot. This would allow to process the images and obtain the position relative to the camera frame in real time for guidance (outer-loop) control. This is feasible as the images can be processed in a reasonable time thanks to the fast algorithms used. Moreover the velocity of the robotic frame can be programmed by estimating the velocity of the robot *a priori* (e.g by simulation). Finally it is worthwhile mentioning that the absolute localisation would not be required to achieve straight-line following in the pool: only the cross-track error between the robot and the path is needed, which can be acquired each time a frame is processed.

2.3.1.5 Third Experiment: Torque Measurement

This experiment aims at measuring the hydrodynamic torque applying at the last joint.

The robot is controlled in inner loop as previously, but this time the desired joint angle associated to the last actuator (actuator number 3, in Figure 2.5) is set to zero so that a proper static torque measurement can be acquired. Indeed, following the datasheet of the Dynamixel, a dynamic torque measurement is not reliable enough and only a static measurement (i.e when the actuator is holding a still position) makes sense. Note also that the built-in torque sensor of the Dynamixel is not outputting a torque measurement directly, but a measure of electric current. Thus, the current-to-torque conversion has to be performed following the discussion in [59].

2.3.1.6 Programming VPM-2D

This section describes briefly the simulation set up (choice of parameters, control law, ...). Practical information such as which functions to be modified and other code-related issues are reported to Appendix A.

The usual parameters are chosen⁶ and recalled here: $a = 0.1\text{m}$, $b = 0.01\text{m}$, $c = 0.01\text{m}$, $\nu = 1\text{e-}6 \text{ m}^2/\text{s}$, $N = 4$.

As the Dynamixel motors implement a PD-type controller, the following control law is used in simulation (obtained heuristically):

$$\tau_i = 500(\phi_i^* - \phi_i) + 50(\dot{\phi}_i^* - \dot{\phi}_i) \quad (2.2)$$

where τ_i is the actuation torque for joint i , ϕ_i is the generalized coordinate associated with the motion of joint i and ϕ_i^* is the desired trajectory (obtained via equation (2.1)). Note that a saturation on the actuation torque is enforced to take into account the limitations of the Dynamixel.

2.3.2 Results

The results of the first experiment are shown in Figures 2.6 and 2.7 for the trajectory following in joint space.

⁶Using the notations introduced in Figure 2.1. Be careful that in the simulator framework, the longitudinal dimension of the ellipse is specified using the major axis $2a$, and not with the major semi-axis a .

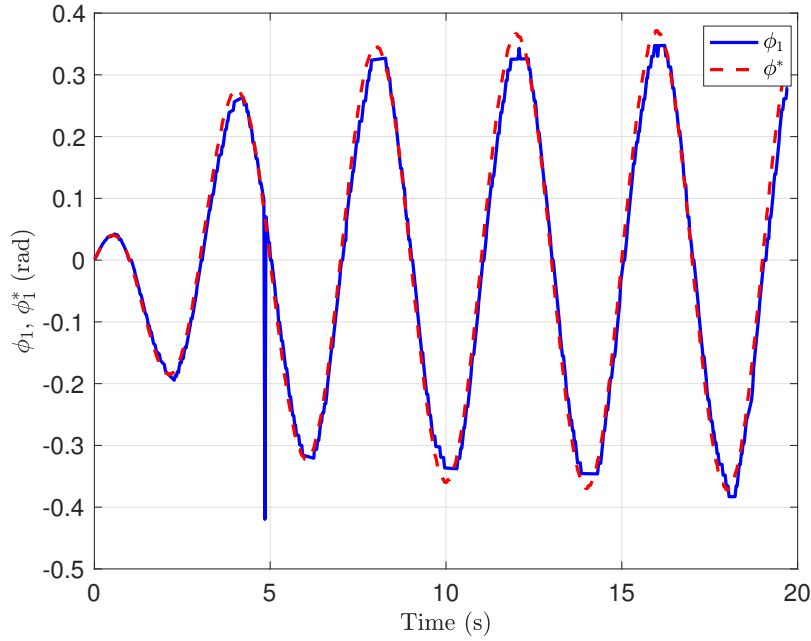


Figure 2.6: Comparison between desired and achieved joint space trajectories for joint number 1.

Figure 2.6 compares the joint desired trajectory issued by the law given in equation (2.1) (dashed red line) with the trajectory achieved by the built-in controller of the Dynamixel (blue solid line) for actuator 1. The other joints presenting similar curves, they are thus not displayed for the sake of conciseness.

The amplitude increases gradually as expected (due to the "gradual-increase" amplitude aforementioned) and reaches its terminal value after about 6 s. Discrepancies between the desired and achieved trajectories are sometimes observed at the peaks with relative errors up to 8%. Very important and sudden discrepancies (like the one arising around 5 s) may be reasonably assimilated to noise in the measurement process.

Sometimes however, discrepancies are more important and the tracking is more chaotic (for example in the presence of disturbances in the water) as illustrated in Figure 2.7 for actuator 2. Brief but important excursions from the desired trajectory are observed which alters the quality of tracking.

The following qualitative observations are also made in this first experiment:

- The observed gait looks like anguilliform motion.
- The robot is not perfectly stable due to the large weight above water (Dynamixel motors), which requires to manually interact with the robot at some point.
- The robot is able to achieve forward motion but with some drift from a straight line, which requires to manually interact with the robot at some point.
- A human presence in the pool is required as it is not possible to interact with the robot from the outside. Consequently, the user has to follow the robot inside the pool, which in turns perturbs the robot and worsen the stability and directionality issues aforementioned.

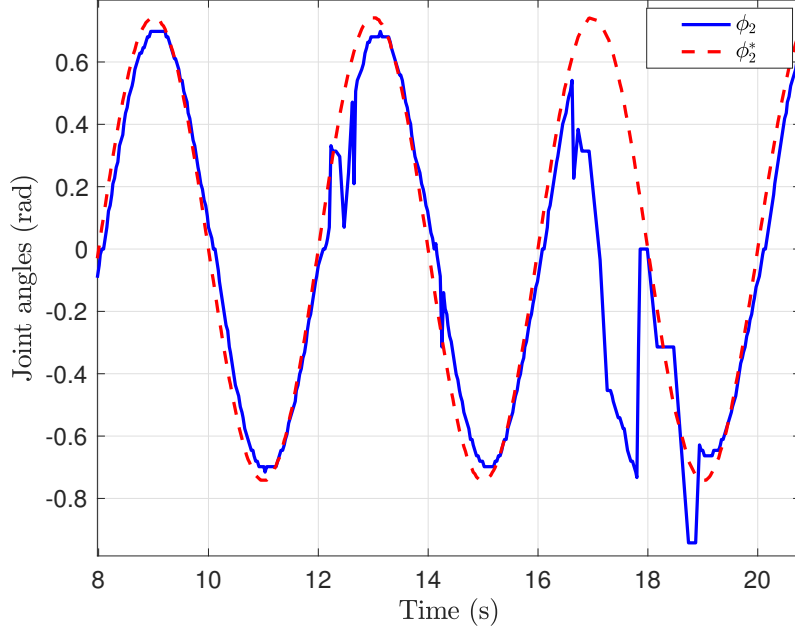


Figure 2.7: Illustration of a discrepancy occurring in the trajectory tracking for joint 2.

- The assembly of all the mechanical parts is easy.
- The ballasting process is a fail and try process: it consists in adding lead balls into the paddle casings of each link and test the buoyancy of the robot in water. Unfortunately, if too many lead balls are added, the robot has to be put out of the water and disassembled to remove some.

The output of the second experiment is presented in Figures 2.8, 2.9 and 2.10.

Some frames obtained via color detection (and DBSCAN) are shown in Figure 2.8. At each time step, the algorithm acquires a frame from the video camera (upper images) and extracts a thresholded image from the original RGB image so that the region considered as "green" appears (lower images). Then the DBSCAN algorithm is applied to eliminate outliers, the centroid of the remaining region is computed and highlighted by a yellow mark.

Figure 2.9 aims at assessing the ability of the color detection algorithm to localize the tail by comparing expected y -axis tail positions over time (inferred from joint angle measurements, i.e odometry) with measured tail positions by color detection. The blue curve (inferred position) is obtained by first acquiring the measured joint angles of all actuators and then performing the required trigonometric computations. The red dots are the successive estimated positions of the tail (centroids of the filtered cluster of green pixels detected in the current image). Errors between the expected and measured amplitudes up to 7cm are observed (corresponding to relative errors up to 15%), along with shifts in time of up to 0.5 s. It should be noted that color detection measurements are not available at each time step: for example around $t = 14$ s a measurement is not available for almost 1s. Note that the time scale in this Figure does not start at zero since the measurements start only when the robot enters the field of view of the video camera, after a few seconds.

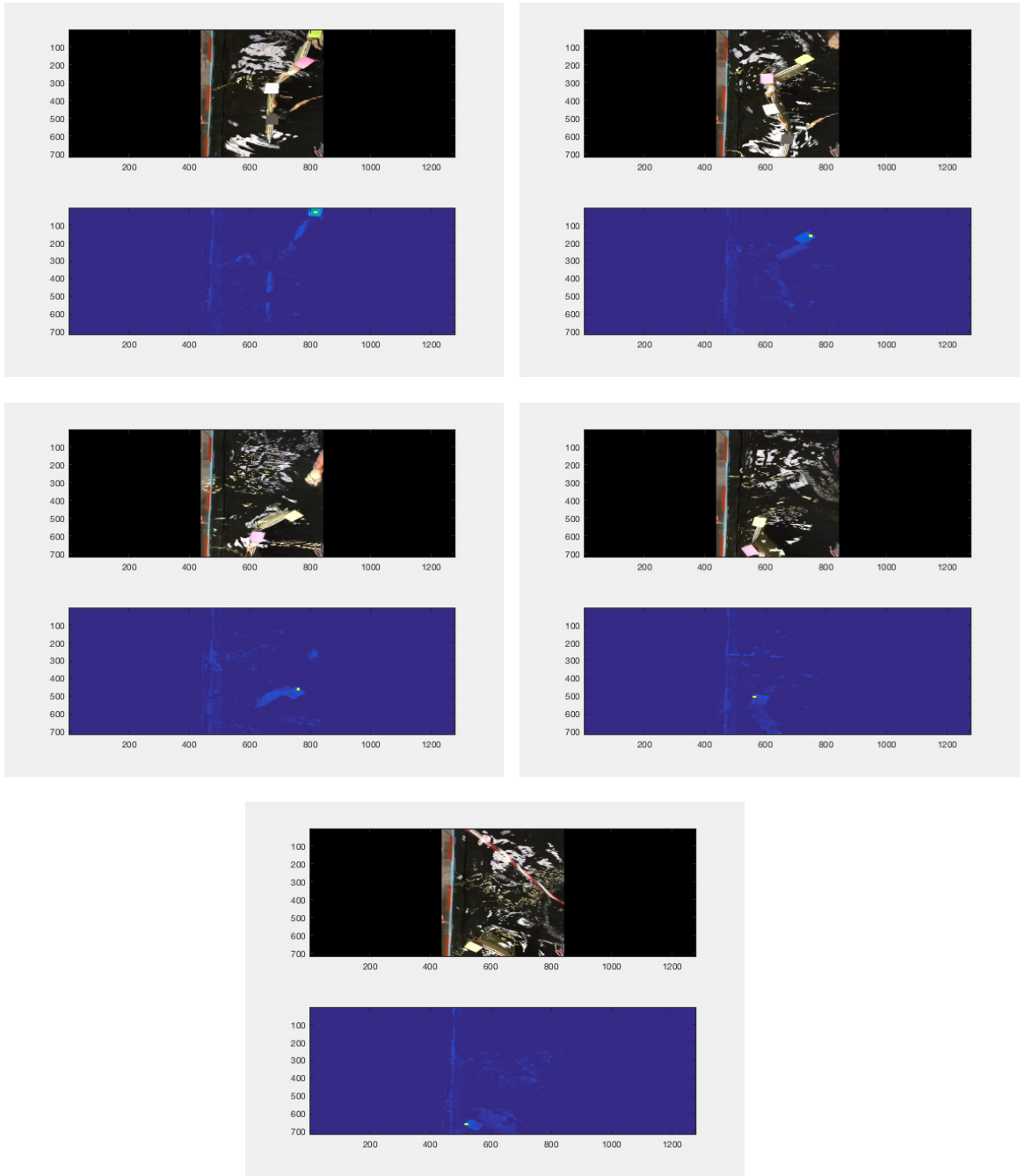


Figure 2.8: Successive frames outputted by the color detection algorithm: the green tag attached to the tail of the robot (upper image) is detected and identified by a yellow marker (lower image)

Figure 2.10 is a comparison between the measured position of the tail (red dots) and the simulation results from VPM-2D with the parameters and controller specified previously. The same qualitative motion pattern is observed for both, with position errors up to 7cm.

The last experiment result is presented in Figure 2.11. In this Figure, the measured and

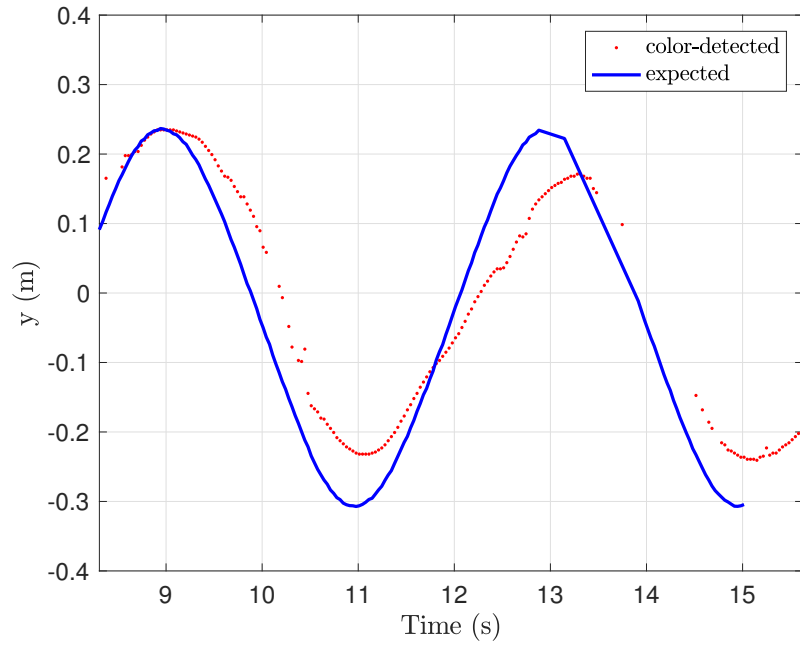


Figure 2.9: Validation of the color detection experiment: comparison between the expected trajectory of the tail (inferred from joint angles measurement) and the measured trajectory with the color detection algorithm.

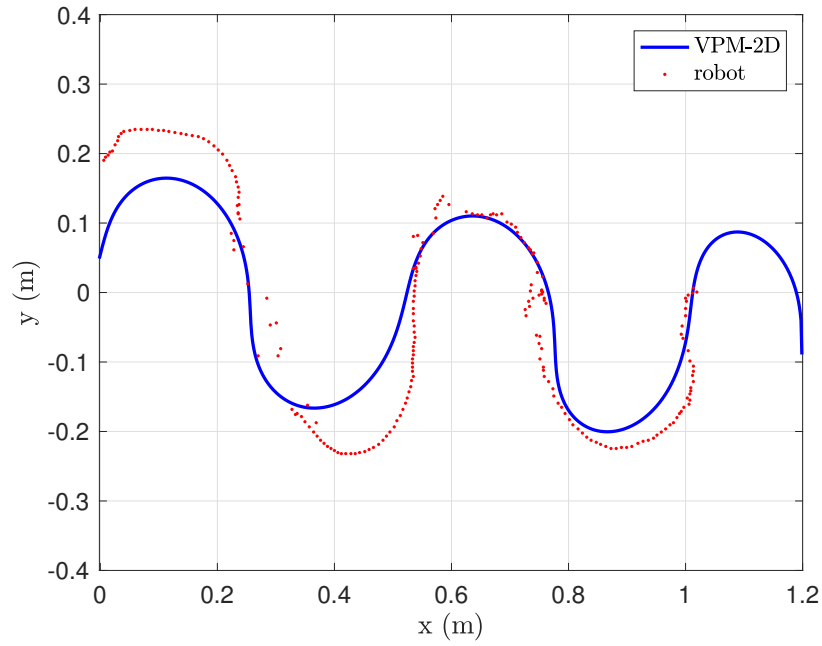


Figure 2.10: Validation of the VPM-2D simulator: comparison between the predicted $x - y$ path of the robot using a VPM-2D simulation and the measured path by color detection of the robot's tail.

the predicted reaction torque at the last joint (static) is plotted versus time. The results fall within the same order of magnitude and both curves have the same oscillatory qualitative

behavior. Nevertheless, a phase shift of approximately 1 s and a (maximum) peak-to-peak differences of 0.2 N.m between the predicted and measured results are observed.

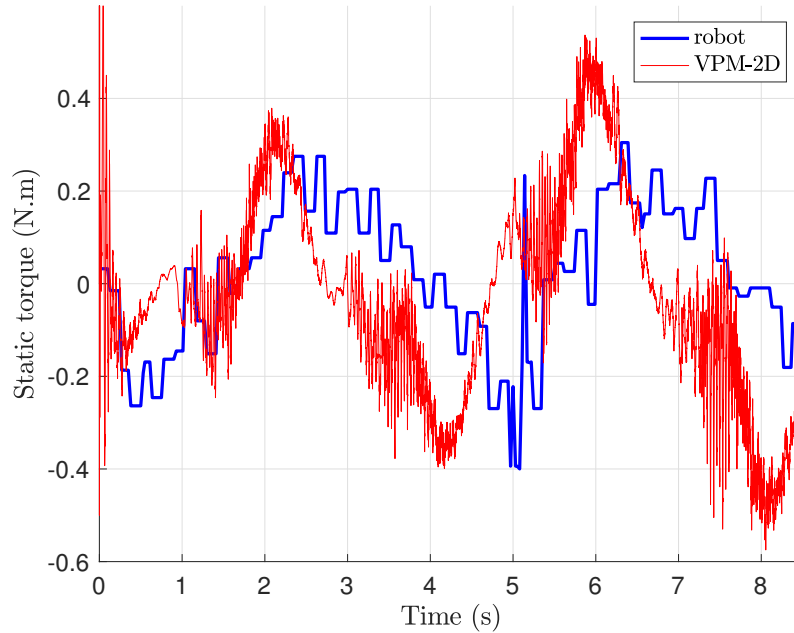


Figure 2.11: Comparison between the predicted static torque (VPM-2D) and the measured static torque at the last joint.

2.3.3 Discussion

The first experiment aims at assessing the ability of the platform to achieve forward motion and follow a prescribed trajectory in joint space. From the results, the capabilities of the Dynamixel motors equipping the platform are acceptable, with a rather good match between the desired and achieved trajectories. However, they are quite sensitive to disturbances in water (arising from the robot itself or by the user walking along in the pool) and wavelets reflected by the pool boundaries. The disturbances cause important local discrepancies regarding the joint-space trajectory tracking, which in turns affects forward motion. Since the outer loop is not closed – i.e there is no guidance control – the robot deviates from a straight-line path. Moreover, stability issues are observed. This has to be corrected by interacting manually with the robot as explained, which in turns causes more disturbances.

Definitely, at this stage, the robot is not able to swim autonomously in the pool without a sporadic human help to stabilize it, but the platform may be kept afloat in the right direction for long enough in order to acquire relevant data for the simulator validation. Further work should be done in order to improve the following aspects:

- **Stability.** Other master's students are currently working on the stability issues by revisiting the robot design: adding a bark-like floating structure to the links. Unfortunately their prototype was not ready yet when the present experiments were carried out.

- Forward motion. Closing the outer loop with a guidance controller is an obvious next step to achieve forward motion. The present work demonstrates the viability of using a combination of color detection with a clustering algorithm DBSCAN for the localization in post-processing. Future work will focus on adapting the set up for a real-time implementation: this includes running the algorithm on an embedded micro-computer, dealing with the motion of the camera frame and implementing a guidance algorithm.
- In order to avoid wavelets reflected at the boundaries, tests should be performed in a larger pool. The width of the pool is actually less than 2 m, which causes important perturbations. Moreover, the dimensions of the pool at this stage prevents testing any guidance algorithm other than straight-line following – and even in that case, an important overshoot may be observed as discussed in Chapter 6.
- Tuning the parameters of the control law. A slowly increasing factor was already added so that the robot starts oscillating smoothly, but the stability may be improved by fine tuning the gait parameters such as the frequency and amplitude of oscillation.
- Torque control. The main drawback of the Dynamixel is its sensitivity to external disturbances. Combined with the impossibility to achieve torque control or use a custom controller, the relevance of this actuator is questionable. Indeed, the user is dependent of the built-in controller, which is not adapted to exploit the knowledge one can have on the environment. A significant improvement would thus to replace the Dynamixel by a torque-controllable actuator and design a controller capable of robustly tracking a trajectory in the presence of disturbances, possibly using real-time information from the environment (model-based control).

The second experiment aims at comparing the simulation-predicted position of the robot with the measured position from a localization system. The localization consists in video processing with feature detection. A color detection algorithm is combined with a clustering algorithm (DBSCAN) to detect the tail of the robot. From this experiment, it is possible to validate the predictions of the simulator.

Indeed, on the one hand a satisfying match is observed between the results of the color detection and the inferred position of the tail from joint angle measurements (odometry). A 7 cm error is observed, which seems reasonable for camera-based localisation and accounts for inevitable deformation of distances with camera images, uncertainties in the exact localization of the tail due to the large size of the color tag (a square of about 4 cm²) or errors in the measurement of the known-length tape in pixel space. Remarkably, thanks to the filtering action of DBSCAN, there seems to be no False Positive⁷ (FP) that would alter the quality of the data. There are however some False Negative⁸ (FN) at some point, probably due to the changing lighting conditions (the robot passes beneath a luminaire which affects the detected RGB image). This is however not a problem even for a real-time implementation as the detection rate stays below 1 s anyways.

On the other hand, the second experiment allows to draw a significant parallel between the simulator predictions and the behavior of the robot. There is indeed a good match

⁷FP: the algorithm mistakenly detects a target that is not the object of interest.

⁸FN: the algorithm fails to detect the object of interest despite present in the frame.

between the measured path and the predicted path, which shows VPM-2D is a good proxy for realistically simulating the motion of anguilliform swimmers. Note that the simulator slightly underestimates the lateral motion of the real process.

The third experiment – hydrodynamic torque measurement at the joint – is a little bit less conclusive in the sense that the match is not as good. A surprising phase shift is observed between the expected and measured results, and the simulator is shown to overestimate the hydrodynamic torque at the joint – which explains why the amplitude of the oscillations predicted by the simulator are smaller than measured. Nevertheless, these discrepancies are mitigated by the fact that the torque sensor of the Dynamixel are not reliable enough as stated by the manufacturer itself in the datasheet [57]. New measurement should be carried out with proper torque sensors, e.g using a strain gauge.

2.4 Conclusion

This Chapter was dedicated to the validation of the VPM-2D simulator with an experiment.

The VPM-2D simulator, along with the robot used in the experiments were first presented.

After that, three experiments were carried out: the first experiment aimed at assessing qualitatively and quantitatively the suitability of the platform to follow a desired trajectory in joint space and achieve forward motion. The results showed satisfying tracking capabilities of the actuators for acquiring relevant data, but highlighted the lack of stability of the platform, drift from straight-line motion, and the sensitivity of the Dynamixel motors to disturbances from the fluid. Some improvements may be done regarding the following aspects: implement an outer-loop controller, implement a custom inner-loop controller with torque-controllable actuators, review the design of the platform regarding the stability.

The second experiment focused on a camera-based localization relying on color detection and pixel clustering (DBSCAN). The localization was successfully validated by comparison with predicted positions from odometry. The information from the localization allowed to validate the VPM-2D simulator by comparison between the detected and simulated paths.

The third experiment was related to static torque measurements at the last joint for comparison with the simulated results. The observations are not conclusive due to the poor reliability of the sensor used, but the results obtained fall within the same order of magnitude.

As a conclusion, this chapter allowed to identify the strengths and the weaknesses of the resources available for this thesis. On the one hand, the VPM-2D simulator captures correctly the behavior of an eel-like swimmer, in particular the robotic platform designed. However, this knowledge comes at the cost of long simulation times (about 12 h for 10 s of simulation in closed inner loop). On the other hand, the robotic platform provides satisfying experimental results for the validation, but suffers from important design issues regarding the stability and the controllability, which prevents its use as an autonomous agent.

The direction chosen for this thesis is to focus on the controllability problem, whereas another team of students will focus on the design aspect (such as solving the stability

issues, replacing the Dynamixel actuators by torque-controllable actuators, and so on).

Acknowledging the limitations of the built-in controller used in the Dynamixel, we seek a robust controller capable of exploiting the information from the environment. Such a model-based approach is thought to overperform the model-free PD-like controller used in the Dynamixel. Moreover, a guidance loop will be added for the purpose of path following. The next Chapters cover the design of these inner- and outer-loop controllers. However, before tackling model-based control, a suitable model of an anguilliform swimmer has to be established. By "suitable", it is intended realistic and simple. Indeed, the VPM-2D simulator already offers a high-fidelity description of the behavior of the system, but it is not a viable solution for control design due to time limitation. The object of the next Chapter is thus to develop a simplified model of an eel in a control design perspective. The VPM-2D simulator, however, will be used throughout this thesis in order to validate the tools and controllers developed.

Chapter 3

Modeling

THIS chapter is an investigation of the simplified dynamics of a N -link eel-like robot, using an inviscid incompressible formulation of Navier-Stokes equations. The motivation arises from the long simulation times experienced with VPM-2D and the desire to obtain an analytic formulation of the dynamics, exploitable for designing a model-based controller. At first, the assumptions made on the fluid and the bodies are listed and discussed. Then, the mathematical formalism used for the treatment of the dynamics and kinematics is presented. The kinematics of a N -link eel is formulated next, along with the configuration and time rate of the kinematics. Finally, the dynamics of the system is established. The approach is inspired from the work of [45], but extends it with the explicit development of the Equations Of Motion (EOM) with Euler-Lagrange formalism.

3.1 Assumptions

First, the various assumptions made for describing the fluid and the bodies (eel) are listed below.

3.1.1 Assumptions on the fluid

- **F1.** Incompressible fluid.
- **F2.** Inviscid fluid.
- **F3.** Irrotational flow.
- **F4.** The volume occupied by the fluid is infinitely large and the fluid is at rest at infinity.
- **F5.** The region occupied by the fluid is topologically connected.

Assumptions **F5** is obviously satisfied and will be used to justify the application of a mathematical result later on. Assumption **F4** is also of practical mathematical use and basically implies that the motion will not be perturbed by external actions (e.g stream, obstacles or wave reflection at the boundaries of the real fluid). If ever these effects occur, which would be the case for any real robot swimming in a pool for example (wave reflection on the walls), they can be treated as small perturbations of the model presently established. **F1**, **F2**, **F3** are important assumptions allowing to simplify and solve for Navier-Stokes

equations. While **F1** is not a strong assumption for a fluid like water which is basically incompressible in normal conditions, **F2** and **F3** are trickier. **F2** says that the fluid can only exert pressure forces on the body (no shear) and **F3** states that the motion of the fluid is entirely due to the solid. However, making these assumptions results in neglecting the important mechanism of vortex shedding¹ that also plays a role in the locomotion of the swimmer.

In Kanso [45], it is suggested that **F3** can still be enforced while taking into account vortex shedding. Indeed, the physics of the mechanism can be treated using "point vortices" that are local singularities of the flow with non-zero vorticity, while the flow is assumed to be irrotational everywhere else [60]. **F2** also still holds as it is pointed out that the vortices generated from the separation of the thin boundary layer of the swimmer are actually negligible compared to other sources of vortex. Kanso's suggestion is thus to use a simplified model based on the perfect-fluid assumption with potential flow, and then possibly add a model of the vorticity on top. As highlighted by Porez in [40], the technique is however computationally expensive which prevents any real-time operation.

In a control design perspective, the simplified model without vortex shedding is thought to capture the dynamics of swim with a sufficient level of fidelity. In other words, we coin that the main propulsive source for eel locomotion is not vortex shedding, but body-to-water transfers of momentum through forces exerted on the fluid. Moreover, we coin that the inviscid assumption, allowing tremendous simplifications, yields a sufficiently realist² model for performing low cost simulations and control design.

3.1.2 Assumptions on the body

- **B1.** The system is neutrally buoyant.
- **B2.** The system consists in a chain of N articulated rigid bodies, denoted $\mathcal{B}_i$.
- **B3.** The bodies are ellipsoidal cylinders of infinite height.
- **B4.** There is no hydrodynamic coupling between links.

Assumptions **B3** and **B1** allow to treat the dynamics as a 2D problem. **B1** is easily achieved, even for the biological fish in agreement with natural observations - it is required that the density of the bodies is equal to the density of the fluid. **B3** justifies the slender shape for the paddles in Chapter 2.

B2 is not true for biological fish (elastic body), but can easily be achieved with robotic fish. **B4** is a simplification that allows to obtain an analytic formulation of Navier-Stokes equations as it will be seen later. It has been shown in [45] that neglecting the influence of other links yields a significant underestimation of the force acting on the bodies. However, taking into account the coupling would imply computing numerically some fluid parameters in real time (using for example costly boundary element methods) which is not acceptable for robotic applications. Nevertheless, it has been reported in [45] that the qualitative

¹Vortex shedding is the emission of vortices subsequent to the separation of the boundary layer around solids moving in a fluid.

²Of course, the model obtained by assuming no viscosity would never achieve the level of fidelity of a VPM-2D simulation. By "sufficiently realist" we mean that the qualitative behavior of the eel can be reproduced with this assumption. This ascertainment will be validated in the next Chapter.

behavior of the motion is still faithfully captured when assuming no coupling. So, in a control-focused perspective, this assumption is acceptable.

In the remainder of this thesis, the aforementioned assumptions are referred by their number to justify mathematical developments or explain discrepancies in simulation results.

3.2 Mathematical formalism

First of all, a mathematical formalism for treating rigorously dynamic problems is introduced, inspired from [61]. The formalism introduces the concept of "vectrix" as a systematic tool to establish vectorial equations and resolve them in various reference frames. Remarkably, vectrices allow to define naturally the rotational operators ("rotation matrices") as elements of the special orthogonal group (SO(3)). In what follows, a systematic distinction will be made between a physical vector and its component in a particular frame. The extra effort will pay off later on when dealing with multiple reference frames.

Let $\mathbb{P}$ be the space of physical vectors (also known first order tensors). To a reference frame $\mathcal{F}_b$, we associate a vectrix defined in terms of its basis vectors:

$$\vec{\mathcal{F}}_b = \begin{bmatrix} \vec{b}^1 \\ \vec{b}^2 \\ \vec{b}^3 \end{bmatrix}, \quad (3.1)$$

where the $\vec{b}^i \in \mathbb{P}$ stand for mutually orthonormal physical vectors called the basis vectors. Any physical vector $\vec{r} \in \mathbb{P}$ can be expressed as a linear combination of the basis vectors of frame $\mathcal{F}_b$, i.e be resolved in frame $\mathcal{F}_b$ as follows:

$$\vec{r} = \vec{\mathcal{F}}_b^T \mathbf{r}_b, \quad (3.2)$$

where $\mathbf{r}_b \in \mathbb{R}^3$ is the component of the physical vector resolved in $\mathcal{F}_b$. A vectrix can thus be viewed as a hybrid object between a vector and a matrix as it carries physical vectors but is subject to the usual operations on matrices (transposition, multiplication, and so on).

Let us define $\vec{T}$ by the linear mapping from $\mathbb{P}$ to $\mathbb{P}$ such that:

$$\vec{T} \cdot \vec{r} \in \mathbb{P}, \quad \forall \vec{r} \in \mathbb{P}, \quad (3.3)$$

i.e $\vec{T}$ is a linear operator turning one physical vector into another. We call $\vec{T}$ a second order tensor or dyadic. A second order tensor can be resolved in an arbitrary frame $\mathcal{F}_b$ as follows:

$$\vec{T} = \vec{\mathcal{F}}_b^T \mathbf{T}_b \vec{\mathcal{F}}_b, \quad (3.4)$$

Let $\mathcal{F}_a$ be another frame defined by its vectrix:

$$\vec{\mathcal{F}}_a = \begin{bmatrix} \vec{a}^1 \\ \vec{a}^2 \\ \vec{a}^3 \end{bmatrix}. \quad (3.5)$$

then, the Direction Cosine Matrix (DCM) between the frame $\mathcal{F}_b$ and the inertial frame $\mathcal{F}_a$ is an element of the special orthogonal group noted $\mathbf{C}_{ab} \in \text{SO}(3)$ and defined by:

$$\mathbf{C}_{ab} = \vec{\mathcal{F}}_a \cdot \vec{\mathcal{F}}_b^\top. \quad (3.6)$$

with, by orthogonality of $\text{SO}(3)$:

$$\mathbf{C}_{ab} \mathbf{C}_{ab}^\top = \mathbf{C}_{ab}^\top \mathbf{C}_{ab} = \mathbf{I}, \quad (3.7)$$

where $\mathbf{I}$ stands for the identity.

Now we tackle the problem of transforming the components of a physical vector from one frame to another. Pre-multiplying equation (3.2) by the vectrix $\vec{\mathcal{F}}_a$, and resolving $\vec{r}$ in frame $\mathcal{F}_a$ yields successively:

$$\vec{\mathcal{F}}_a \vec{r} = \vec{\mathcal{F}}_a \vec{\mathcal{F}}_b^\top \mathbf{r}_b, \quad (3.8)$$

$$\vec{\mathcal{F}}_a \vec{\mathcal{F}}_a^\top \mathbf{r}_a = \vec{\mathcal{F}}_a \vec{\mathcal{F}}_b^\top \mathbf{r}_b, \quad (3.9)$$

Noting that $\vec{\mathcal{F}}_a \vec{\mathcal{F}}_a^\top = \mathbf{I}$ by (3.7) and using the definition of the DCM in (3.6), we have:

$$\mathbf{r}_a = \mathbf{C}_{ab} \mathbf{r}_b, \quad (3.10)$$

which shows how any vector resolved in frame $\mathcal{F}_b$ can be resolved in frame $\mathcal{F}_a$ via the DCM.

Conversely, we have:

$$\mathbf{r}_b = \mathbf{C}_{ab}^\top \mathbf{r}_a, \quad (3.11)$$

Similarly, any second order tensor resolved in frame $\mathcal{F}_b$ can be resolved in frame $\mathcal{F}_a$ via the suitable transformation. Respectively pre- and post-multiplying equation (3.4) by $\vec{\mathcal{F}}_a$ and $\vec{\mathcal{F}}_a^\top$, and using the definition of the DCM in (3.6) yields:

$$\vec{\mathcal{F}}_a \vec{T} \vec{\mathcal{F}}_a^\top = \vec{\mathcal{F}}_a \vec{\mathcal{F}}_b^\top \mathbf{T}_b \vec{\mathcal{F}}_b \vec{\mathcal{F}}_a^\top, \quad (3.12)$$

$$\mathbf{T}_a = \mathbf{C}_{ab} \mathbf{T}_b \mathbf{C}_{ab}^\top, \quad (3.13)$$

which gives the desired transformation.

Finally we introduce the skew-symmetric operator $\cdot^\times$ as the mapping from $\mathbb{R}^3$ to the Lie algebra of $\text{SO}(3)$, defined by the relation : $\mathbf{v} \times \mathbf{u} = \mathbf{v}^\times \mathbf{u}$, $\forall \mathbf{v}, \mathbf{u} \in \mathbb{R}^3$. In particular, defining $\boldsymbol{\omega}_a^{ba}$ the angular velocity of frame $\mathcal{F}_b$ with respect to frame $\mathcal{F}_a$ and resolved in $\mathcal{F}_a$, the derivative of the DCM is given by [62]:

$$\dot{\mathbf{C}}_{ab} = \boldsymbol{\omega}_a^{ba \times} \mathbf{C}_{ab}. \quad (3.14)$$

If moreover the DCM reduces to a simple rotation of angle θ about an axis defined by $\mathbf{k}_a$, then the following holds [62]:

$$\frac{d}{d\theta} \mathbf{C}_{ab} = \mathbf{k}_a^\times \mathbf{C}_{ab}. \quad (3.15)$$

3.3 Kinematics

This section introduces the kinematic equations of the system, describing the geometry of motion without any concern for the cause of motion. Referring to Figure 3.1 representing a N -segment eel-like robot (multibody system in open chain), let $\mathcal{F}_a$ be the absolute or inertial frame³ defined by the dextral orthonormal basis of physical vectors $\vec{a}^1, \vec{a}^2, \vec{a}^3$ ($\vec{a}^3$ pointing outside the plane), let $\mathcal{F}_{b_i}$ be the frame attached to the i -th body, defined by the dextral orthonormal basis of physical vectors $\vec{b}_i^1, \vec{b}_i^2, \vec{b}_i^3$, and let us introduce the vector of generalized coordinates⁴ $\mathbf{q} \in \mathbb{R}^n, n = N + 2$:

$$\mathbf{q} = \begin{bmatrix} \mathbf{q}_b \\ \mathbf{q}_s \end{bmatrix}, \quad (3.16)$$

with $\mathbf{q}_b = [x \ y \ \theta]^\top$ the so-called *body variables* or *group variables* and $\mathbf{q}_s = [\phi_1 \ \phi_2 \ \dots \ \phi_{N-1}]^\top$ the *shape variables*. The body variables are not directly controllable and describe the pose of the first segment; the shape variables are directly controllable via actuator torques and consist in the joint relative⁵ angles (counted positively counterclockwise).

The geometric characteristics of a segment are: the major semi-axis of the ellipse a , the minor semi-axis of the ellipse b , the tip-to-joint distance c . The length $L = 2l = 2a + 2c$ is also used abundantly in what follows.

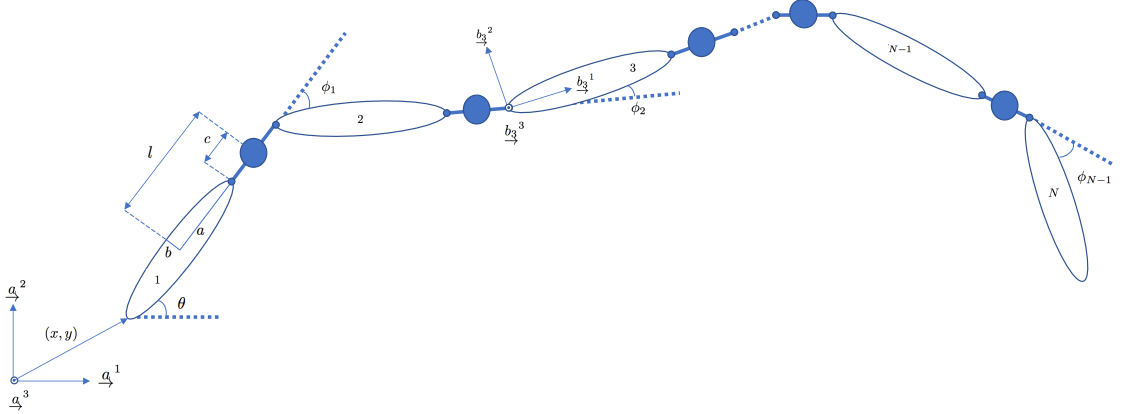


Figure 3.1: N -segment eel-like robot kinematic.

From the Figure above, we can define the DCM $\mathbf{C}_{ab_i}$ that characterizes a z -axis rotation

³In the remainder of this thesis, the axis oriented along $\vec{a}^1, \vec{a}^2, \vec{a}^3$ will be referenced as " x -axis", " y -axis" and " z -axis", respectively.

⁴There are indeed $n = N + 2$ degrees of freedom. The system being constrained to planar motion (because of the neutrally buoyant assumption), the first segment has 3 degrees of freedom while there is only 1 degree of freedom in rotation per body for the remaining $N - 1$ segments.

⁵For the sake of conciseness, the denomination "relative" is dropped in the remainder of this thesis.

of angle $\mathcal{Q}(i) = \theta + \sum_{\delta=1}^{i-1} \phi_{\delta}$ from $\mathcal{F}_{b_i}$ to $\mathcal{F}_a$:

$$\mathbf{C}_{ab_i} = \begin{bmatrix} \cos(\mathcal{Q}(i)) & -\sin(\mathcal{Q}(i)) & 0 \\ \sin(\mathcal{Q}(i)) & \cos(\mathcal{Q}(i)) & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (3.17)$$

Let $\vec{v}_i$ and $\vec{\omega}_i$ denote⁶ respectively the linear and angular velocity vectors of (the center of mass of) body i ; we call "twist" of body i the following augmented vector:

$$\vec{\xi}_i = \begin{bmatrix} \vec{v}_i \\ \vec{\omega}_i \end{bmatrix}. \quad (3.18)$$

Resolving the twist in inertial frame yields:

$$\xi_{i_a} = \tilde{\mathbf{C}}_{ab_i} \xi_{i_{b_i}} = \begin{bmatrix} \mathbf{C}_{ab_i} \mathbf{v}_{i_{b_i}} \\ \mathbf{C}_{ab_i} \boldsymbol{\omega}_{i_{b_i}}^{ba} \end{bmatrix} = \begin{bmatrix} \mathbf{v}_{i_a} \\ \boldsymbol{\omega}_{i_a}^{ba} \end{bmatrix}, \quad (3.19)$$

where $\tilde{\mathbf{C}}_{ab_i}$ stands for the augmented DCM defined by:

$$\tilde{\mathbf{C}}_{ab_i} = \begin{bmatrix} \mathbf{C}_{ab_i} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{C}_{ab_i} \end{bmatrix}. \quad (3.20)$$

Now let us introduce $\mathbf{r}_{i_a}(\mathbf{q}) \in \mathbb{R}^6$ the generalized-coordinate-dependant pose of body i resolved in inertial frame, such that $\xi_{i_a} = \dot{\mathbf{r}}_{i_a}$. Application of the chain rule to the latter equation yields:

$$\xi_{i_a} = \dot{\mathbf{r}}_{i_a} = \sum_{k=1}^n \frac{\partial \mathbf{r}_{i_a}}{\partial q_k} \dot{q}_k = \mathbf{J}_i(\mathbf{q}) \dot{\mathbf{q}}, \quad (3.21)$$

where $\mathbf{J}_i$ is identified as the Jacobian of body i by analogy with robotic manipulators. It represents the relation between the generalized coordinates and the twist of link i resolved in inertial frame.

The Jacobian for the N -link eel-like robot presented in Figure 3.1 can be obtained by an explicit evaluation of formula (3.21). The detailed computation of the Jacobian, along with the Jacobian transpose and Jacobian derivative is deferred to Appendix B.

3.4 Dynamics

In this section, the EOM are derived using a Lagrangian formulation. The expression of the Lagrangian is obtained by considering the whole system body and fluid following a method reported by Lamb in [63] and dating back from Kirchhoff's work when he developed a theory of solids moving in potential flows. Following Kirchhoff's method, the obtained Lagrangian of the whole system can be reduced to the kinetic energy of the body along with the contribution of the fluid via the concept of "added inertia": intuitively, when moving, the body carries a portion of the fluid which can be modeled by an addition in the inertia of the body.

⁶Pay attention to the subscript " i ": it is used to characterize a body- i -related quantity, not to say that this quantity is resolved in the i -th body-attached frame. By contrast, the notation " b_i " used as a subscript for the component of a vector means that this vector has been resolved in body- i frame

This idea of augmenting the mass of the body, though really intuitive, can actually be shown to derive rigorously from an analytic resolution of the simplified Navier-Stokes equations using the assumptions on the fluid and the body. This is the object of this section.

First, the Lagrangian of the whole system fluid and body is given by:

$$\mathcal{L} = \sum_{i=1}^N \mathcal{T}_{\mathcal{B}_i} + \mathcal{T}_f, \quad (3.22)$$

where $\mathcal{T}_{\mathcal{B}_i}$ and $\mathcal{T}_f$ stand for the kinetic energy of the i -th body and the fluid respectively:

$$\mathcal{T}_{\mathcal{B}_i} = \frac{1}{2} \vec{\xi}_i \cdot \vec{M}_i \cdot \vec{\xi}_i, \quad (3.23)$$

$$\mathcal{T}_f = \frac{1}{2} \int_V \rho \|\vec{u}\|_2^2 dv, \quad (3.24)$$

where $\vec{M}_i$ is the augmented tensor (stacked just as vectrices are) obtained by assembling the mass and second-order moment of inertia tensors associated with the i -th body, ρ is the density of the fluid, $\vec{u}$ is the velocity vector field of the fluid and the integration is performed over the volume V of the fluid. First let us note that the integrand $\rho \|\vec{u}\|_2^2 \in \mathbb{L}_2$ since, from **F4**, the velocity vanishes at infinity. So the integral is bounded and the infinite volume fluid has a finite energy.

In equation (3.24), $\vec{u}$ satisfies the Navier-Stokes equations, given here in their vorticity formulation:

$$\begin{aligned} \frac{\partial \vec{\Omega}}{\partial t} + (\vec{u} \cdot \nabla) \vec{\Omega} &= (\vec{\Omega} \cdot \nabla) \vec{u} - \vec{\Omega} (\nabla \cdot \vec{u}) \\ &+ \frac{1}{\rho^2} \nabla \rho \times \nabla p + \nabla \times \left(\frac{\nabla \cdot \vec{\tau}}{\rho} \right) + \nabla \times \left(\frac{\vec{B}}{\rho} \right), \end{aligned} \quad (3.25)$$

and:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \vec{u}) = 0, \quad (3.26)$$

where $\vec{\Omega} = \nabla \times \vec{u}$ is the vorticity of a fluid particle, $\vec{B}$ accounts for the sum of external forces acting on the body, $\vec{\tau}$ is the viscous stress tensor, p is the local pressure.

It is widely known that there (probably) exists no closed form solution of the equations above in the general case. However, using the hypotheses discussed in section 3.1, it is possible to make the following simplifications:

On the one hand, by **F1**, the continuity equation (3.26) reduces to:

$$\nabla \cdot \vec{u} = 0. \quad (3.27)$$

On the other hand, by **F3**, the vorticity is zero which implies that:

$$\nabla \times \vec{u} = 0 \quad (3.28)$$

Thus, from equation (3.28) and using **F5**, there exists a well defined potential flow ϕ , such that the velocity $\vec{u}$ derives from that potential:

$$\vec{u} = \nabla\phi. \quad (3.29)$$

Replacing in equation (3.27) yields the well known Laplace equation:

$$\Delta\phi = 0. \quad (3.30)$$

This stands for a degenerated form of the Navier-Stokes equations, for which an analytic solution can be sought. Kirchhoff showed that it is possible to express the scalar potential ϕ as a function of the geometry and twist of the submerged body only [63]. This intuition is a direct consequence of **F3**: the motion of the fluid is entirely due to that of the body. Consequently it can be shown that the fluid velocity vector along the normal $\vec{n}_i$ of body i boundary $\partial\mathcal{B}_i$ at any point is equal to the normal velocity of the surface at that point:

$$\nabla\phi \cdot \vec{n}_i = (\vec{\omega}_i \times \vec{r} + \vec{v}_i) \cdot \vec{n}_i, \quad (3.31)$$

where ω_i and $\vec{v}_i$ are given by the twist equation (3.18) and $\vec{r}$ is the position vector of an arbitrary point on the boundary of the body. From there, we conclude that ϕ has the desired form involving the vectorial potential functions $\vec{\chi}_i$ and $\vec{\varphi}_i$ depending only on body i geometry:

$$\vec{\phi} = \sum_{i=1}^N \vec{\omega}_i \cdot \vec{\chi}_i + \vec{v}_i \cdot \vec{\varphi}_i, \quad (3.32)$$

the latter equation satisfying Laplace equation (3.30) with the following Neumann conditions:

$$\nabla\vec{\varphi}_i \cdot \vec{n}_i = \vec{n}_i, \quad (3.33)$$

$$\nabla\vec{\chi}_i \cdot \vec{n}_i = \vec{r} \times \vec{n}_i. \quad (3.34)$$

Now, using the following remarkable identity:

$$\nabla \cdot (\phi \nabla \phi) = \phi \Delta \phi + \nabla \phi \cdot \nabla \phi, \quad (3.35)$$

along with the potential formulation of the velocity in equation (3.29) and Laplace equation (3.30), one can successively rewrite the fluid kinetic energy in (3.24) as:

$$\begin{aligned} \mathcal{T}_f &= \frac{1}{2} \int_V \rho \|\vec{u}\|_2^2 dv \\ &= \frac{1}{2} \int_V \rho \nabla \phi \cdot \nabla \phi dv \\ &= \frac{1}{2} \int_V \rho \nabla \cdot (\phi \nabla \phi) dv \end{aligned} \quad (3.36)$$

Applying Green's theorem:

$$\mathcal{T}_f = -\frac{1}{2} \int_{\sum_{i=1}^N \partial\mathcal{B}_i} \rho \phi \nabla \phi \cdot \vec{n}_i dS \quad (3.37)$$

where the minus sign is due to the orientation of the normal to the i -th body surface $\vec{n}_i$, pointing inward the fluid. Note also that the integral term corresponding to the boundary of the domain at infinity is zero, in agreement with **F4**.

Finally, applying Kirchhoff's hypothesis in (3.32):

$$\begin{aligned} \mathcal{T}_f = & \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \vec{\omega}_i \overrightarrow{X_{ij}} \vec{\omega}_j + \vec{v}_i \overrightarrow{\Xi_{ij}} \vec{\omega}_j \\ & + \vec{\omega}_i \overrightarrow{\Xi_{ij}} \vec{v}_j + \vec{v}_i \overrightarrow{\Phi_{ij}} \vec{v}_j, \end{aligned} \quad (3.38)$$

with $\overrightarrow{X_{ij}}$, $\overrightarrow{\Xi_{ij}}$ and $\overrightarrow{\Phi_{ij}}$ being the added masses/inertia associated respectively with rotational motion, coupled rotational-translational motion and translational motion, defined by⁷:

$$\overrightarrow{X_{ij}} = \rho \int_{\partial \mathcal{B}_i} \vec{\chi}_i \otimes \nabla \cdot \vec{\chi}_j \vec{n} \, dS, \quad (3.39)$$

$$\overrightarrow{\Xi_{ij}} = \rho \int_{\partial \mathcal{B}_i} \vec{\chi}_i \otimes \nabla \cdot \vec{\varphi}_j \vec{n} \, dS, \quad (3.40)$$

$$\overrightarrow{\Phi_{ij}} = \rho \int_{\partial \mathcal{B}_i} \vec{\varphi}_i \otimes \nabla \cdot \vec{\varphi}_j \vec{n} \, dS. \quad (3.41)$$

Note that the integral over the N bodies boundaries in equation eq:dev3 is written as the sum of the integrals over each boundaries, then the integral and summation operators have been permuted ; this is mathematically consistent as the integrand is Lebesgue-integrable (following from **F4**).

Two major simplifications are now made:

- First, we consider no coupling between rotational motion and translational motion thanks to the two symmetry axes of the ellispoidal shape of the bodies and the 2D nature of the problem (**B3**). Consequently $\overrightarrow{\Xi_{ij}}$ vanishes.
- From **B4**, the influence of body i on body j is assumed negligible, resulting in $\overrightarrow{X_{ij}}$ and $\overrightarrow{\Phi_{ij}}$ to vanish for $i \neq j$. The validity of this hypothesis is discussed in [45].

In other words, the kinetic energy can be rewritten as:

$$\begin{aligned} \mathcal{T}_f &= \frac{1}{2} \sum_{i=1}^N \vec{\omega}_i \overrightarrow{X_{ii}} \vec{\omega}_i + \vec{v}_i \overrightarrow{\Phi_{ii}} \vec{v}_i \\ &= \frac{1}{2} \sum_{i=1}^N \vec{\xi}_i \overrightarrow{M_{ii}^a} \vec{\xi}_i, \end{aligned} \quad (3.42)$$

where $\overrightarrow{M_{ii}^a}$ is the augmented tensor of added inertia and resolves in body i frame into a diagonal matrix depending only on the geometry.

⁷The operation $\otimes$ stands for the dyadic product between two first order tensors, generating a second order tensor.

It is worthwhile mentioning that the fluid-related kinetic energy obtained in (3.42) is now expressed in terms of the twist (body quantities), and can be directly added to the body kinetic energy in equation (3.23) to form the total Lagrangian of the system:

$$\mathcal{L} = \frac{1}{2} \sum_{i=1}^N \vec{\xi}_i \cdot (\vec{M}_{ii}^a + \vec{M}_i) \cdot \vec{\xi}_i. \quad (3.43)$$

In [64], Newman gives a way of computing the body-resolved tensor of added inertia for various geometries. It is thus tempting to resolve in body frame the physical vectors and tensors present in the Lagrangian as follows:

$$\mathcal{L} = \frac{1}{2} \sum_{i=1}^N \xi_{ib}^T \underbrace{(\mathbf{M}_{iib_i}^a + \mathbf{M}_{ib_i})}_{\mathcal{D}_{ib_i}} \xi_{ib}, \quad (3.44)$$

where $\mathcal{D}_{ib_i}$ is the so-called body-resolved (configuration independent) wet inertia matrix. For an ellipsoidal geometry (**B3**), and dropping the index i as all bodies are (geometrically) the same: $\mathcal{D}_{b11} = \rho\pi b^2 + m$, $\mathcal{D}_{b22} = \rho\pi a^2 + m$, $\mathcal{D}_{b66} = \frac{1}{8}\rho\pi(a^2 - b^2)^2 + \frac{1}{4}m(a^2 + b^2)$, all other components zero.

The above formulation in terms of body-resolved quantities is useful for computing the wet inertia matrix. However, in order to use Euler-Lagrange formalism, a formulation in terms of the generalized coordinates is sought. Thus, using the transformation from $\mathcal{F}_{b_i}$ to $\mathcal{F}_a$ given by equation (3.11), and invoking the generalized coordinates via (3.21) yields:

$$\begin{aligned} \mathcal{L} &= \frac{1}{2} \sum_{i=1}^N \xi_{ia}^T \tilde{\mathbf{C}}_{ab_i} \mathcal{D}_{ib_i} \tilde{\mathbf{C}}_{ab_i}^T \xi_{ia}, \\ &= \frac{1}{2} \sum_{i=1}^N \dot{\mathbf{q}}^T \mathbf{J}_i^T \mathcal{D}_{ia} \mathbf{J}_i \dot{\mathbf{q}}, \end{aligned} \quad (3.45)$$

where the absolute-frame-resolved wet inertia matrix is: $\mathcal{D}_{ia} = \mathcal{M}_{iia}^a + \mathcal{M}_{ia} = \tilde{\mathbf{C}}_{ab_i} \mathbf{M}_{iib_i}^a \tilde{\mathbf{C}}_{ab_i}^T + \mathbf{M}_{ib_i}$ ($\mathbf{M}_{ib_i}$ is invariant under z -axis rotations). It is of paramount importance to stress that this matrix is configuration dependant, through the added inertia matrix. In other words, the added inertia of the body varies with the generalized coordinates.

Introducing the generalized Inertia matrix:

$$\mathbf{D} = \sum_{i=1}^N \mathbf{J}_i^T \mathcal{D}_{ia} \mathbf{J}_i, \quad (3.46)$$

the Lagrangian can be expressed more compactly in matrix form:

$$\mathcal{L} = \frac{1}{2} \dot{\mathbf{q}}^T \mathbf{D} \dot{\mathbf{q}}, \quad (3.47)$$

The Euler-Lagrange equations can now be used to establish the dynamics:

$$\frac{d}{dt} \left(\frac{\partial \mathcal{L}}{\partial \dot{\mathbf{q}}} \right) - \frac{\partial \mathcal{L}}{\partial \mathbf{q}} = \mathbf{Q} \quad (3.48)$$

where $\mathbf{Q}$ is the $\mathbb{R}^{N+2}$ generalized force vector given by:

$$\mathbf{Q} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \boldsymbol{\tau} \end{bmatrix}, \quad (3.49)$$

with $\boldsymbol{\tau} = [\tau_1 \ \tau_2 \ \dots \ \tau_{N-1}]^\top$ the joint torque vector.

Applying Euler-Lagrange yields the EOM:

$$\mathbf{D}\ddot{\mathbf{q}} + \left(\dot{\mathbf{D}} - \frac{1}{2} \left(\frac{\partial \mathbf{D}}{\partial \mathbf{q}} \dot{\mathbf{q}} \right)^\top \right) \dot{\mathbf{q}} = \mathbf{Q} \quad (3.50)$$

where the term in bracket can be identified as the Coriolis matrix $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) = \dot{\mathbf{D}} - \frac{1}{2} \left(\frac{\partial \mathbf{D}}{\partial \mathbf{q}} \dot{\mathbf{q}} \right)^\top$. Note that the term $\frac{\partial \mathbf{D}}{\partial \mathbf{q}} \dot{\mathbf{q}}$ defines a matrix whose j -th column is the vector: $\frac{\partial \mathbf{D}}{\partial q_j} \dot{\mathbf{q}}$. Also, the dependency on the torque input in the generalized force vector can be shown via $\mathbf{Q} = \mathbf{H}\boldsymbol{\tau}$, with:

$$\mathbf{H} = \begin{bmatrix} \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times N-1} \\ \mathbf{0}_{N-1 \times 3} & \mathbf{I}_{N-1 \times N-1} \end{bmatrix} \quad (3.51)$$

This formulation of the EOM will be useful for future theoretical analysis. However, it is not suitable for a computer implementation for which it is desirable to have componentwise expressions of the Inertia and Coriolis matrices. For that purpose, the scalar form of the Euler-Lagrange's equations may be used:

$$\sum_{j=1}^n D_{kj} \ddot{q}_j + \sum_{l=1}^n \sum_{j=1}^n c_{ljk} \dot{q}_l \dot{q}_j = Q_k, \quad (3.52)$$

where the Christoffel symbols are defined by:

$$c_{ljk} = \frac{1}{2} \left(\frac{\partial D_{kj}}{\partial q_l} + \frac{\partial D_{kl}}{\partial q_j} - \frac{\partial D_{lj}}{\partial q_k} \right) \quad (3.53)$$

The derivative terms can be obtained by an explicit derivation of the Inertia matrix:

$$\frac{\partial \mathbf{D}}{\partial q_m} = \sum_{i=1}^N \frac{\partial \mathbf{J}_i^\top}{\partial q_m} \mathcal{D}_{i_a} \mathbf{J}_i + \mathbf{J}_i^\top \frac{\partial \mathcal{D}_{i_a}}{\partial q_m} \mathbf{J}_i + \mathbf{J}_i^\top \mathcal{D}_{i_a} \frac{\partial \mathbf{J}_i}{\partial q_m}. \quad (3.54)$$

with:

$$\begin{aligned} \frac{\partial \mathcal{D}_{i_a}}{\partial q_m} &= \frac{\partial \mathcal{M}_{ii_a}^a}{\partial q_m} \\ &= \frac{\partial \tilde{\mathbf{C}}_{ab_i}}{\partial q_m} \mathbf{M}_{ii_{b_i}}^a \tilde{\mathbf{C}}_{ab_i}^\top + \tilde{\mathbf{C}}_{ab_i} \mathbf{M}_{ii_{b_i}}^a \frac{\partial \tilde{\mathbf{C}}_{ab_i}^\top}{\partial q_m} \\ &= \frac{\partial \mathcal{Q}(i)}{\partial q_m} \left(\tilde{\mathbf{a}}_a^{3 \times} \mathcal{M}_{ii_a}^a + \mathcal{M}_{ii_a}^a \tilde{\mathbf{a}}_a^{3 \times \top} \right) \\ &= \frac{\partial \mathcal{Q}(i)}{\partial q_m} \left(\tilde{\mathbf{a}}_a^{3 \times} \mathcal{M}_{ii_a}^a - \mathcal{M}_{ii_a}^a \tilde{\mathbf{a}}_a^{3 \times} \right) \end{aligned} \quad (3.55)$$

where $\tilde{\mathbf{a}}_a^{3 \times} = \text{diag}(\mathbf{a}_a^{3 \times}, \mathbf{a}_a^{3 \times})$ and $\mathbf{a}_a^3 = [0 \ 0 \ 1]^\top$ is the component of the vertical axis resolved in inertial frame. Note that $\cdot^\times$ stands for the skew symmetric operator. The Jacobian, Jacobian transpose and Jacobian derivative are given in Appendix B.

This achieves the development of the EOM of the N links eel-like robot presented in Figure 3.1.

3.5 Conclusion

This Chapter is concerned with the development of a simplified dynamic model for an anguilliform swimmer in a perfect fluid. The simplifying assumptions made in this Chapter allow to treat the hydrodynamic efforts as an addition to the inertia of the body. This allows to combine the Lagrangian of the fluid and the body into a form that depends exclusively on body configuration-related variables (twist).

This approach is inspired from the work of Kanso [45]. However, contrary to Kanso, an explicit derivation of the kinematics (see Appendix B) and the EOM using Euler-Lagrange formalism is proposed here.

Indeed, to the author's knowledge, neither in Kanso's paper, nor in any posterior literature implementing Kanso's approach, were the EOM explicitly established in the Euler-Lagrange formalism, as done here. Indeed, Kanso's approach consists in establishing the Lagrangian of the system, then applying the so-called "Lagrange-Poincare reduction", method whose mathematical formulation heavily relies on group theory, geometrical mechanics and algebraic topology.

By contrast in this thesis, the Lagrange-Poincare part is bypassed and, starting from the Lagrangian, the classical and widely-known Euler-Lagrange approach is used. The main contribution of this Chapter is thus a different formulation of Kanso's work, maybe more suitable for the control engineering community as it does not require heavy mathematical background⁸.

The interest of the model presented in this Chapter is twofold. On the one hand, it offers an analytic formulation of the hydrodynamic efforts, in terms of the system configuration and geometry only (via the added inertia). Such an analytic formulation can be used in a control design perspective, as will be the case in Chapter 5. On the other hand, the model derived here can be used to develop a simulation environment suitable for quickly testing controllers at no cost, by contrast to the VPM-2D simulator, much more computationally intensive and time-consuming.

The question now arising concerns the reliability of such a simplified model. Indeed, the perfect fluid and irrotationality are strong assumptions that cannot be asserted carelessly as they may lead to physical non-sense. The next Chapter thus addresses the question of the validity of the presented model.

⁸Note however that many authors from the control community use the concept of added inertia in their model, without explicitly referring to the perfect fluid and irrotationality assumptions that motivate its introduction.

Chapter 4

Model Validation

The model derived in Chapter 3 is used to develop a new simulation tool for rapidly designing and testing controllers for eel-like swimmers. The so-called "inviscid simulator" (by contrast with the "viscous-flow" VPM-2D simulator) is validated against literature results (from [45] using the same hypotheses) and compared to viscous-flow simulators, in particular VPM-2D. Finally, energy checks are conducted to verify if the model satisfies to the conservation of mechanical energy.

4.1 Simulation Results

The model developed in the previous Chapter was implemented on MATLAB / SIMULINK, with the possibility to specify any desired number of links. In a model validation perspective, we simulate a $N = 3$ links robot since it is the benchmark commonly used in the literature to compare results. At each time step, the inertia and Coriolis matrices established in the previous section are assembled and the acceleration at time step k given by:

$$\ddot{\mathbf{q}}_k = \mathbf{D}_k^{-1}(\boldsymbol{\tau}_k - \mathbf{C}_k \dot{\mathbf{q}}_k) \quad (4.1)$$

is integrated with an explicit (4,5) Runge-Kutta method.

In the study of locomotion of dynamical systems, it is common to enforce a specified kinematic pattern at the joints and simulate the evolution of the other variables. The following prescribed trajectory in joint space has been studied in the literature (for example: [45], [56]) and will serve as a benchmark for testing the forward motion of the system:

$$\phi_1 = -\cos\left(t - \frac{\pi}{2}\right), \quad (4.2)$$

$$\phi_2 = -\cos(t). \quad (4.3)$$

The parameters of the system are chosen as follows (the reader is referred to Figure 3.1 for the notations):

- ellipse aspect ratio $\frac{a}{b} = 10$
- tip-to-joint distance: $c = 0.2a$
- via **B1**, the density of the fluid is equal to the density of the body: $\rho = \rho_b$.

- the characteristic length $L = 2l = 2a + 2c$ is used for nondimensionalization.

The simulation is launched with the following initial conditions: $\mathbf{q} = [0 \ 0 \ 0 \ 0 \ -1]^\top$ and $\dot{\mathbf{q}} = [0 \ 0 \ 0 \ 0 \ 0]^\top$. Figure 4.1 shows the prescribed joint trajectories enforced by the sinusoidal pattern (4.2) and (4.3). Figure 4.2 shows the body variable evolution (namely x, y and θ – pose of the first segment, referring to Figure 3.1) over time. The Figure shows that forward motion is achieved with the characteristic oscillatory motion pattern of an eel. The angle θ is oscillating around a constant value of -0.5 rad, which implies that the eel moves forward in a straight direction oriented south-east, as expected from the observed evolution of x and y .

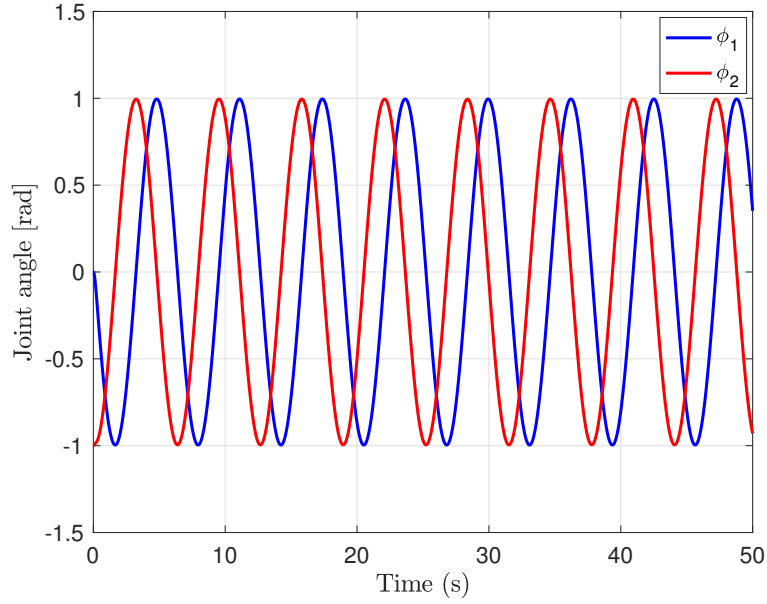


Figure 4.1: Prescribed kinematic pattern at the joints (controllable or shape variables)

In order to validate the model against results from the literature, the trajectory of the second link's centroid is first nondimensionalized by the characteristic length L , then reported in the x, y -plane of Figure 4.3 (green curve). A comparison is done with the trajectories obtained in the same conditions for various simulators in the literature: VPM-2D (blue), Kanso's inviscid model [45] (black) and Eldredge's viscous solver [65] (red).

The Figure shows the match between the present work and Kanso's model (the discrepancy only comes from the interpolation of the data by hand from Kanso's paper), which demonstrates the validity of the present computer implementation of the model and also the equivalence between the Euler-Lagrange and Lagrange-Poincare methods for establishing the EOM, see the conclusive remark of Chapter 3.

The comparison between the inviscid and viscous models shows a qualitatively similar behavior. The discrepancy between the viscous and inviscid models arises mainly from the unmodeled viscous dynamics in the inviscid model: the inviscid model is faster as there is no viscous forces (proportional to velocity) acting on it. Remarkably, the match between the viscous and inviscid models is pretty decent at the beginning: this is because the velocity of the "viscous swimmer" is low, and so are the viscous forces acting on it.

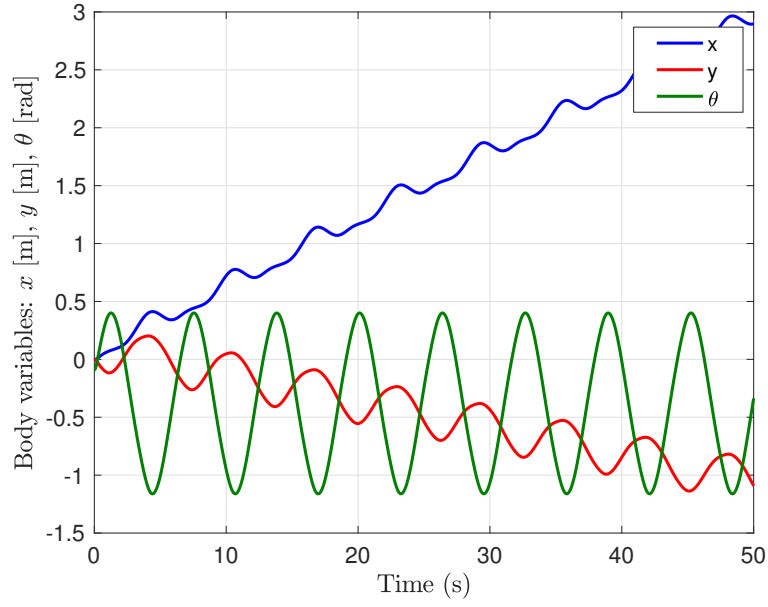


Figure 4.2: Trajectory in the body variable space.

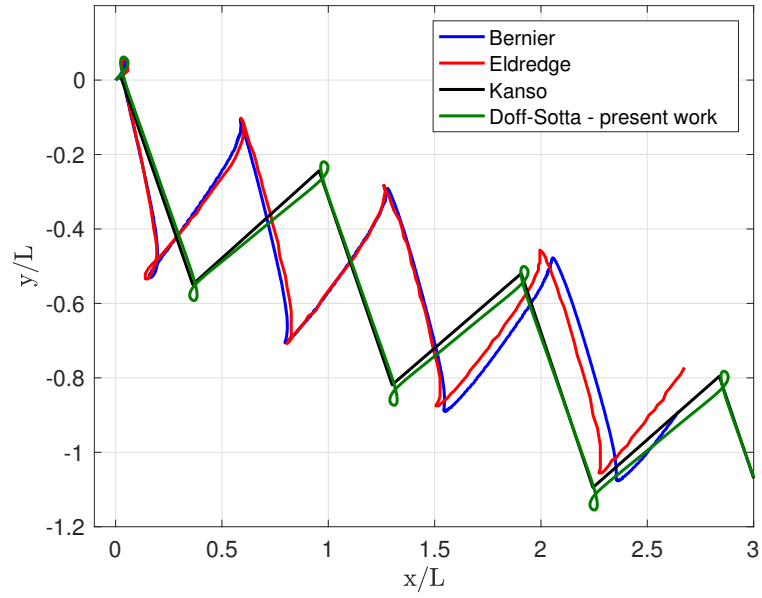


Figure 4.3: Nondimensionalized trajectory of the central body centroid. In blue and red: viscous-based models ; in green and black: inviscid Kirchhoff theory model.

However, as the viscous swimmer gets faster, the viscous forces are more important and the discrepancy is higher.

4.2 Energy check

Remarkably, the kinetic energy of the system, that may be identified to the Lagrangian in equation (3.47), only posses quadratic terms in $\dot{\mathbf{q}}$. In the most general case however, the positions of the particles composing a system not only depend on the generalized coordinates, but may also depend explicitly on time. In that case, the total kinetic energy of a system $\mathcal{T}$ is summed up by the contribution due to a quadratic function of $\dot{\mathbf{q}}$ noted $\mathcal{T}_2$, a linear function of $\dot{\mathbf{q}}$ noted $\mathcal{T}_1$ and $\mathcal{T}_0$, not a function of $\dot{\mathbf{q}}$. Using this notation, the Hamiltonian usually defined in terms of the generalized momenta p_i :

$$\mathcal{H} = \sum_{i=1}^n p_i \dot{q}_i - \mathcal{L}, \quad (4.4)$$

can be written more compactly as:

$$\mathcal{H} = \mathcal{T}_2 - \mathcal{T}_0 + \mathcal{V}, \quad (4.5)$$

where $\mathcal{V}$ is the potential energy of the system.

In our case we thus have ($\mathcal{V} = 0$ and $\mathcal{T}_0 = 0$):

$$\mathcal{H} = \mathcal{T}_2 = \mathcal{T} = \mathcal{E}_{mech}, \quad (4.6)$$

where $\mathcal{E}_{mech}$ is the total mechanical energy. This shows the equality between the mechanical energy and the Hamiltonian.

Taking the derivative of the Hamiltonian given by equation (4.4), it is easily shown using Euler-Lagrange equations that:

$$\dot{\mathcal{H}} = \sum_{i=1}^n Q_i \dot{q}_i - \frac{\partial \mathcal{L}}{\partial t}. \quad (4.7)$$

From equations (4.7) and (4.6), we conclude that the total mechanical energy is conserved if no torque is applied to the system via the actuators. Indeed, because there is no explicit dependence of the Lagrangian on time and no generalized forces applied to the system, the Hamiltonian is thus constant (and so is the mechanical energy).

The conservation of the total mechanical energy can be verified numerically by simulation. The following experiment is done: the system is given an initial velocity ($\dot{\mathbf{q}} = [0 \ 0 \ 0.2 \ 0 \ 0]^T$) and no torque is applied at the joint, i.e the system passively reacts to its environment (whose effect is included by nature in the dynamics of the system via the body-fluid Lagrangian). Figure 4.4 shows the evolution of the total kinetic energy over time. As expected from the energetic analysis above, the total mechanical energy is conserved in the absence of non-conservative forces such as actuator torques (and viscous forces – not taken into account in the model).

Note however that if joint torques are applied, the assumption on the non-conservative forces does not hold, and the Hamiltonian derivative (4.7) is not zero anymore:

$$\dot{\mathcal{H}} = \sum_{i=1}^n Q_i \dot{q}_i \neq 0. \quad (4.8)$$

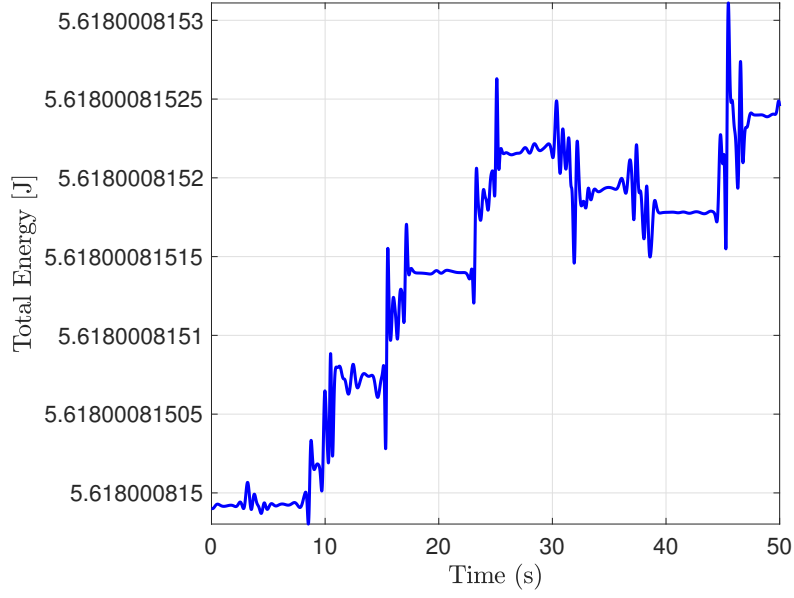


Figure 4.4: Conservation of the total mechanical energy in the absence of non-conservative forces.

Hence, the total mechanical energy is not conserved anymore as seen in Figure 4.5 showing the evolution of the mechanical energy subsequently to the application of a sinusoidal torque pattern in joint space.

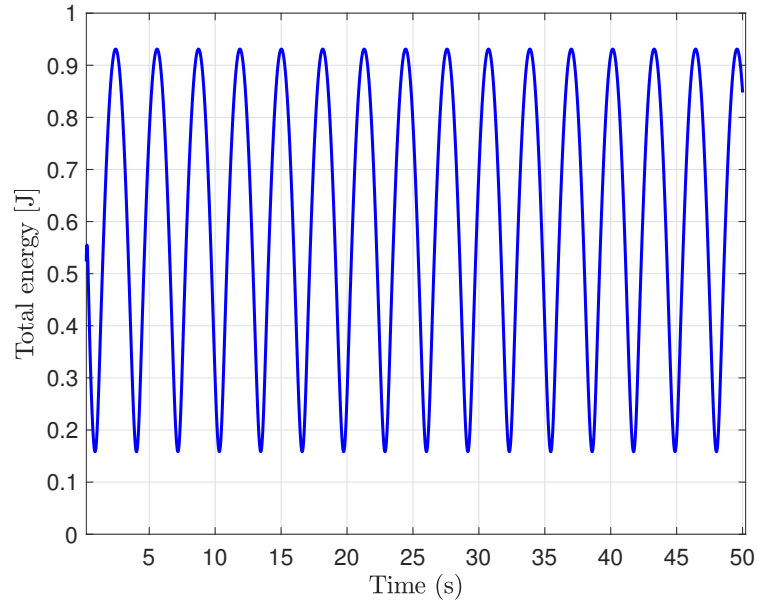


Figure 4.5: Variation of the total mechanical energy under actuator torques.

4.3 Conclusion

The model developed in the previous Chapter is simulated and compared with results from the literature.

In particular, simulations are confirming the prediction, following an energy check, that the total mechanical energy is conserved under some assumptions.

The most important result of this Chapter is the comparison of simulations with other models found in the literature. On the one hand, the model achieves the same trajectory as Kanso's swimmer in the same conditions, which validates the inviscid simulator presented in this work. On the other hand, the comparison with the viscous flow simulators shows a qualitative match between models.

Obviously, the inviscid simulator is not able to capture the full nonlinearities in the fluid as VPM-2D does. Moreover, it will never replace a high-fidelity solver in the context of CFD analysis. However, it was shown in this Chapter to predict the locomotion of an eel-like swimmer to a satisfying-enough extent. Thus, from now on, the inviscid simulator will be regarded as a good proxy for the swimming process and will be extensively used in the next Chapter in order to design and test a model-based controller.

Chapter 5

Passivity-based Robust Control

A robust control approach based on the passivity properties of the system is implemented in this Chapter. The Chapter starts by a review of fundamental concepts in Input-Output (I - O) stability theory, passivity theory and the mathematical theory of positive real matrices: the concept of $\mathbb{L}_2$ -stability, along with the important passivity theorem and Kalman-Yakubovich-Popov (KYP) lemma are presented. Then, the passivity properties of the system are highlighted: the I-O map between actuator torques and joint angle velocities is shown to be passive. In agreement with the strong form of the passivity theorem, a negative feedback interconnection of a passive system with a Very Strictly Passive (VSP) controller yields $\mathbb{L}_2$ -stability of the closed loop. Thus, a VSP controller is sought in the form of a (linear) Strictly Positive Real (SPR) transfer function, synthesized in an optimal way using a Linear Quadratic Regulator (LQR) approach and KYP lemma. The present work also extends an existing design technique to underactuated systems. After that, the negative feedback interconnection of the plant and the SPR controller is achieved, and tested in simulation for both the inviscid simulator and VPM-2D. Several types of disturbances are introduced in the system to assess the robustness of the control approach, and the results are shown to outperform classical PD approach for joint inner control. Finally the limitations of the SPR robust control approach are identified in simulation and some potential extensions are proposed.

5.1 Mathematical Foundations of I-O Stability Theory, Passivity Theory and Positive Real Matrices

This section is not an exhaustive introduction to passivity-based control. The reader is directed to [66] for a detailed presentation.

The concept of Input-Output (I-O) stability is motivated by the limitations of Lyapunov's stability theory. If Lyapunov's stability theory offers a variety of tools to characterize the convergence of nonlinear systems trajectories submitted to initial conditions, it says nothing on what happens in the presence of external inputs such as noise or disturbances¹. In a robust-control design perspective, it is thus desirable to use a theory describing the input-output map of a dynamical system, rather than its stability with respect to an equilibrium point given an initial condition [67]. I-O stability theory offers a mathematical

¹It does not mean that Lyapunov's theory is invalidated by I-O stability theory, nor that the latter extends the former strictly speaking. Instead, they are completing each other, and Lyapunov stability theory will even be used later on in this Chapter without contradicting or interfering with I-O stability theory.

formalism to characterize the stability and response of a dynamical system submitted to "sufficiently bounded" signals.

The term "sufficiently bounded" is first clarified by the following set of definitions [66], [67], [68].

Definition 1 (Square Integral Functions Space $\mathbb{L}_2$). *The space $\mathbb{L}_2$ is the space of all square integral functions defined by:*

$$\mathbb{L}_2 := \left\{ \mathbf{u} : \mathbb{R}^+ \rightarrow \mathbb{R}^m \mid \int_0^\infty \mathbf{u}^T(t) \mathbf{u}(t) dt < \infty \right\}, \quad (5.1)$$

where $\mathbf{u}$ is an arbitrary function. The $\mathbb{L}_2$ space is a Hilbert space on which the norm is induced by the inner product:

$$\langle \mathbf{u}, \mathbf{v} \rangle := \int_0^\infty \mathbf{u}^T(t) \mathbf{v}(t) dt, \quad \|\mathbf{u}\|_2 := \sqrt{\langle \mathbf{u}, \mathbf{u} \rangle}, \quad (5.2)$$

where $\mathbf{u} \in \mathbb{L}_2, \mathbf{v} \in \mathbb{L}_2, \langle \cdot, \cdot \rangle$ is the inner product, and $\|\cdot\|_2$ the $\mathbb{L}_2$ -norm.

Definition 2 (Truncation of a Function). *The truncation of a function $\mathbf{u}$, denoted $\mathbf{u}_T : \mathbb{R}^+ \rightarrow \mathbb{R}^m, T \in \mathbb{R}^+$ is defined by:*

$$\mathbf{u}_T(t) = \begin{cases} \mathbf{u}(t) & 0 \leq t \leq T \\ \mathbf{0} & t > T \end{cases} \quad (5.3)$$

Definition 3 (Extended $\mathbb{L}_2$ Space). *The extended $\mathbb{L}_2$ space is defined by:*

$$\mathbb{L}_{2e} := \left\{ \mathbf{u} : \mathbb{R}^+ \rightarrow \mathbb{R}^m \mid \mathbf{u}_T(t) \in \mathbb{L}_2, \forall T \in \mathbb{R} \right\}, \quad (5.4)$$

where $\mathbb{L}_2 \subset \mathbb{L}_{2e}$. The $\mathbb{L}_{2e}$ space is a Hilbert space on which the truncated norm:

$$\|\mathbf{u}_T\|_2 = \sqrt{\int_0^\infty \mathbf{u}_T^T(t) \mathbf{u}_T(t) dt} = \sqrt{\int_0^T \mathbf{u}^T(t) \mathbf{u}(t) dt} =: \|\mathbf{u}\|_{2T}, \quad (5.5)$$

is induced by the truncated inner product:

$$\langle \mathbf{u}_T, \mathbf{v}_T \rangle := \int_0^\infty \mathbf{u}_T^T(t) \mathbf{v}_T(t) dt = \int_0^T \mathbf{u}^T(t) \mathbf{v}(t) dt = \langle \mathbf{u}, \mathbf{v} \rangle_T, \quad (5.6)$$

where $\mathbf{u} \in \mathbb{L}_{2e}$ and $\mathbf{v} \in \mathbb{L}_{2e}$

Signals that are in $\mathbb{L}_2$ can be thought as "finite" energy signals, or "sufficiently bounded" signals. The space $\mathbb{L}_{2e}$ extends $\mathbb{L}_2$, allowing for the use of a wider class of "sufficiently bounded" functions, as shown in the following examples.

Example 1. *The function $f(t) = e^t \sin(t) \notin \mathbb{L}_2$. However, $f(t) = e^t \sin(t) \in \mathbb{L}_{2e}$*

This second example is more practical and show how the extended $\mathbb{L}_{2e}$ space includes "everyday-life" signals like a step of constant magnitude:

Example 2. *Consider the Heaviside function (step of constant magnitude a) $\forall t \in \mathbb{R}^+, H(t) = a$. Obviously, $H(t) \notin \mathbb{L}_2$, however $H(t) \in \mathbb{L}_{2e}$.*

The definition of an I-O-stable or $\mathbb{L}_2$ -stable system can now be precised [67]:

Definition 4 ($\mathbb{L}_2$ -stability). *A system represented by the operator $\mathcal{G} : \mathbb{L}_{2e}(U) \rightarrow \mathbb{L}_{2e}(Y)$ (I-O representation), mapping the inputs $\mathbf{u} \in \mathbb{L}_{2e}(U)$ to the outputs $\mathbf{y} \in \mathbb{L}_{2e}(Y)$ via $\mathbf{y}(t) = (\mathcal{G}\mathbf{u})(t)$, is said to be Input-Output $\mathbb{L}_2$ -stable, or simply $\mathbb{L}_2$ -stable if,*

$$\mathbf{u} \in \mathbb{L}_2(U) \rightarrow \mathbf{y} \in \mathbb{L}_2(Y) \quad (5.7)$$

The concept of I-O stability can thus be summed-up by the following key idea: a system is I-O stable if "finite energy in yields finite energy out" [67]. As an example, a pendulum would be $\mathbb{L}_2$ -stable with respect to its equilibrium position as any finite input brings it back to its equilibrium position. However an inverted pendulum is not $\mathbb{L}_2$ -stable *per se* as any finite-energy input pushes it away from its equilibrium position.

Within the context of I-O stability, two powerful results have been around since the 1980's: the small gain theorem (a fundamental tool, notably used in $\mathcal{H}_\infty$ -loop robust control) and the passivity theorem. The latter will be of particular interest in this Chapter as it offers a way of achieving robust and stable closed loop for systems presenting the so-called "passivity" property.

Intuitively, a dynamical system is passive if it only dissipates energy without producing any. The term originates from circuit theory where a passive network is made of resistors, capacitors and inductors. Using an electromechanical analogy, a dynamical system is passive if it can be reduced to the three fundamental mechanical elements: mass, spring, damper. It is no coincidence that engineers pioneering at the pre-computer era were simulating the behavior of dynamical systems by experimentation on equivalent electrical circuits.

Evaluating the performances of an electric circuit (e.g stability) is usually done by applying test signals at some point (input) and measuring the response at some other point (output). Likewise, the concept of passivity for mechanical system only makes sense when defined with respect to a given I-O map (hence its right place in I-O stability theory). Usually, in a dynamical system, passivity arises for collocated quantities like forces and velocities or torque and angular rates.

More formally, the passivity of a system with respect to a given input-output map is defined mathematically by [67]:

Definition 5 (Passivity). *A square system $\mathbf{y}(t) = (\mathcal{G}\mathbf{u})(t)$, $\mathcal{G} : \mathbb{L}_{2e}(U) \rightarrow \mathbb{L}_{2e}(Y)$ is Very Strictly Passive (VSP) if there exists real constant $\delta > 0$, $\epsilon > 0$ and β such that:*

$$\langle \mathbf{u}, \mathbf{y} \rangle_T \geq \delta \|\mathbf{u}\|_{2T}^2 + \epsilon \|\mathbf{y}\|_{2T}^2 + \beta, \quad \forall \mathbf{u} \in \mathbb{L}_{2e}(U), \quad \forall T \in \mathbb{R}^+ \quad (5.8)$$

In particular, the system is:

- *Input Strictly Passive (ISP) when the above equation holds with $\delta > 0$ and $\epsilon = 0$*
- *Output Strictly Passive (OSP) when the above equation holds with $\delta = 0$ and $\epsilon > 0$*
- *passive when equation holds with $\delta = \epsilon = 0$*

The above definition states that a system is passive with respect to a given input-output map if the scalar product of its input-output pair is bounded from below [69]. For collocated input-output map such as force-velocity, voltage-current or torque-angular rate, the product represents the instantaneous power externally supplied by the system and the inner product (integral over time) represents the energy. This energy is thus bounded and the bound is

a measure of the total finite energy available in the system for a given initial condition. The passivity property thus states that there is a finite limit on the energy that can be extracted in the system. [70]. The VSP-ness of a system stands for a stronger form of passivity.

The passivity theorem can now be stated [67], [66]:

Theorem 1 (Passivity Theorem). *Consider the negative feedback interconnection of $\mathcal{G}_1 : \mathbb{L}_{2e} \rightarrow \mathbb{L}_{2e}$ and $\mathcal{G}_2 : \mathbb{L}_{2e} : \mathbb{L}_{2e} \rightarrow \mathbb{L}_{2e}$ shown in Figure 5.1. If $\mathcal{G}_1$ is passive and $\mathcal{G}_2$ is VSP, the negative feedback interconnection is $\mathbb{L}_2$ -stable.*

Proof. see [67] using the dissipativity formalism (a generalization of passivity for non-square systems) or [71]. $\square$

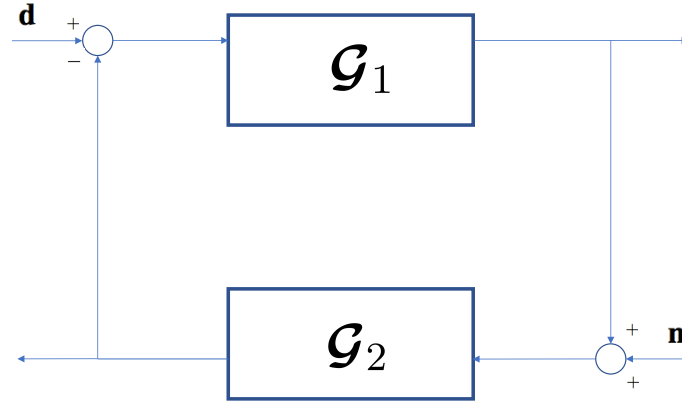


Figure 5.1: Feedback interconnection of a passive system with a VSP system. Passivity theorem states that the closed loop is input-output stable. The signals $\mathbf{d}$ and $\mathbf{n}$ represent external inputs (e.g disturbances and noise).

The passivity theorem states in particular that if a system $\mathcal{G}_1$ is passive with respect to an I-O map, it is possible to design a VSP controller $\mathcal{G}_2$ stabilizing the closed loop in the sense of I-O stability. In other words, such a feedback interconnection would be robust to finite $\mathbb{L}_2$ -norm external input (disturbance, noise). The passivity theorem is thus of paramount importance in robust control theory: it is a practical tool for achieving $\mathbb{L}_2$ -stable interconnections of passive/VSP systems. For example it tells which signals should be fed-back and fed-in. It also tells which type of controller should be used. However, the passivity theorem does not tell how to obtain a particular type of controller, say a VSP.

A result known as the Kalman-Yakubovich-Popov lemma provides a practical criterion for designing VSP controllers. But before, we need some technical results related to the theory of positive real matrices [67]:

Definition 6 (Positive Real Transfer Matrix). *A rational transfer matrix $\mathbf{G}(s) \in \mathbb{C}^{m \times m}$ is Positive Real (PR) if:*

- all elements of $\mathbf{G}(s)$ are analytic in $\text{Re}\{s\} > 0$, and

- $\mathbf{G}^*(s) + \mathbf{G}(s) \geq 0$ in $\text{Re}\{s\} > 0$, or equivalently:
 - the poles on the imaginary axis are simple and have non-negative residues, and
 - $\mathbf{G}^*(j\omega) + \mathbf{G}(j\omega) \geq 0, \forall \omega \in \mathbb{R}$ with $j\omega$ not a pole of any element of $\mathbf{G}(j\omega)$.

Definition 7 (Strictly Positive Real Transfer Matrix). *A stable rational transfer matrix $\mathbf{G}(s) \in \mathbb{C}^{m \times m}$ is Strictly Positive Real (SPR) if $\mathbf{G}(s - \delta)$ is PR for some $\delta > 0$; that is, if:*

- all elements of $\mathbf{G}(s)$ are analytic $\text{Re}\{s\} \geq 0$,
- $\mathbf{G}^*(j\omega) + \mathbf{G}(j\omega) > 0, \omega \in \mathbb{R}$,
- $\mathbf{Z} = \mathbf{G}^\top(\infty) + \mathbf{G}(\infty) \geq 0$, or $\lim_{\omega \rightarrow \infty} \omega^2 [\mathbf{G}^*(j\omega) + \mathbf{G}(j\omega)] > 0$ if $\mathbf{Z}$ is singular.

Geometrically these definitions mean that the transfer function is mapping the closed right half of the complex plane to the closed right half of the $\mathbf{G}(s)$ plane. The definitions are motivated by circuit theory where passive networks are represented by a (strictly) positive real rational transfer function [72].

We are now in position to state the KYP lemma [72]:

Lemma 1 (Kalman-Yakubovich-Popov). *Let an observable and controllable LTI system whose state-space realization is given in standard form:*

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}, \quad (5.9)$$

$$\mathbf{y} = \mathbf{C}\mathbf{x} + \mathbf{D}\mathbf{u}. \quad (5.10)$$

The system above is Strictly Positive Real (SPR) iff there exists real matrices $\mathbf{P} = \mathbf{P}^\top > 0$ and $\mathbf{Q} = \mathbf{Q}^\top > 0$ such that

$$\mathbf{P}\mathbf{A} + \mathbf{A}^\top \mathbf{P} = -\mathbf{Q}, \quad (5.11)$$

$$\mathbf{P}\mathbf{B} = \mathbf{C}^\top, \quad (5.12)$$

Proof. see [71]. □

This result is also often referenced in the literature as the "positive real" lemma.

An important corollary of this lemma, related to Linear Time-Invariant (LTI) systems, states as follows:

Lemma 2 (VSP-ness of LTI SPR Systems). *Let a LTI system whose state-space realization is given in standard form:*

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}, \quad (5.13)$$

$$\mathbf{y} = \mathbf{C}\mathbf{x} + \mathbf{D}\mathbf{u}. \quad (5.14)$$

If the triple $(\mathbf{A}, \mathbf{B}, \mathbf{C})$ is SPR and the matrix $\mathbf{D} = \mathbf{D}^\top$, then the system is VSP.

Proof. If the triple $(\mathbf{A}, \mathbf{B}, \mathbf{C})$ is SPR, it satisfies KYP lemma. Let us consider the energy-like function:

$$\mathcal{V} = \frac{1}{2} \mathbf{x}^\top \mathbf{P} \mathbf{x}. \quad (5.15)$$

Deriving with respect to time and expanding yields:

$$\begin{aligned} \dot{\mathcal{V}} &= \mathbf{x}^\top \mathbf{P} \dot{\mathbf{x}}, \\ &= \mathbf{x}^\top \mathbf{P} \mathbf{A} \mathbf{x} + \mathbf{x}^\top \mathbf{P} \mathbf{B} \mathbf{u}. \end{aligned} \quad (5.16)$$

By virtue of KYP lemma:

$$\begin{aligned} \dot{\mathcal{V}} &= -\frac{1}{2} \mathbf{x}^\top \mathbf{Q} \mathbf{x} + \mathbf{C} \mathbf{x}^\top \mathbf{u}, \\ &= -\frac{1}{2} \mathbf{x}^\top \mathbf{Q} \mathbf{x} + \mathbf{y}^\top \mathbf{u} - \mathbf{u}^\top \mathbf{D}^\top \mathbf{u}. \end{aligned} \quad (5.17)$$

Now using the fact that the $\mathbf{Q}$ and $\mathbf{D}$ matrices are symmetric, the weighted norms associated with these matrices can (in particular) be bounded from below by their minimum real eigenvalues $\underline{\lambda}$, as a consequence of the spectral theorem:

$$\dot{\mathcal{V}} \leq -\frac{1}{2} \underline{\lambda}(\mathbf{Q}) \mathbf{x}^\top \mathbf{x} + \mathbf{y}^\top \mathbf{u} - \underline{\lambda}(\mathbf{D}) \mathbf{u}^\top \mathbf{u}. \quad (5.18)$$

Integrating both side from 0 to T :

$$\mathcal{V}(T) - \mathcal{V}(0) \leq -\frac{1}{2} \underline{\lambda}(\mathbf{Q}) \|\mathbf{x}\|_T^2 + \langle \mathbf{y}, \mathbf{u} \rangle_T - \underline{\lambda}(\mathbf{D}) \|\mathbf{u}\|_T^2 \quad (5.19)$$

and the system is thus VSP:

$$-\mathcal{V}(0) + \frac{1}{2} \underline{\lambda}(\mathbf{Q}) \|\mathbf{x}\|_T^2 + \underline{\lambda}(\mathbf{D}) \|\mathbf{u}\|_T^2 \leq \langle \mathbf{y}, \mathbf{u} \rangle_T. \quad (5.20)$$

□

This important result draws a parallel between VSP and SPR systems: if one can find a LTI system whose transfer function is SPR, lemma 2 states that the system is also VSP. The KYP lemma then provides a practical way of designing a SPR LTI. The problem of designing a VSP controller now reduces to finding a SPR controller, which is a significantly simpler task via the KYP lemma.

To put in a nutshell, this section introduced the benefits of I-O stability theory for robust control design. In particular, the concept of passivity is central and the passivity theorem guarantees $\mathbb{L}_2$ -stability for feedback interconnections of passive and VSP systems. A LTI system is then a good candidate for easily designing a VSP controller: KYP lemma offers a practical way of making it SPR, and lemma 2 guarantees it is VSP. But before tackling the problem of designing a passivity-based robust controller, it remains to prove that the plant to control is passive. This is the object of the next section.

5.2 Passivity

The previous section introduced a theoretical frame to robustly control a passive system. In the context of this thesis, it is desirable to apply these techniques to an eel-like robot. The following questions now naturally arise: "Is an eel-like robot passive with respect to

some controllable I-O map ?" Or, more precisely, "Is a *simplified* model of the eel (e.g the inviscid model) passive with respect to some controllable I-O map ?"

This section provides an answer by highlighting the passivity properties of the system.

We first start by showing a technical result that will be of valuable use in the demonstration of the passivity:

Lemma 3. *The matrix $\boldsymbol{\eta} = \dot{\mathbf{D}} - 2\mathbf{C}$, where $\mathbf{D}$ and $\mathbf{C}$ are defined in Chapter 3, is skew-symmetric.*

Reasoning componentwise:

$$\begin{aligned}\eta_{\gamma\beta} &= \dot{D}_{\gamma\beta} - 2C_{\gamma\beta} = \sum_{\alpha=1}^n \frac{\partial D_{\gamma\beta}}{\partial q_{\alpha}} \dot{q}_{\alpha} - \sum_{\alpha=1}^n \left(\frac{\partial D_{\gamma\beta}}{\partial q_{\alpha}} + \frac{\partial D_{\gamma\alpha}}{\partial q_{\beta}} - \frac{\partial D_{\alpha\beta}}{\partial q_{\gamma}} \right) \dot{q}_{\alpha} \\ &= \sum_{\alpha=1}^n \left(\frac{\partial D_{\alpha\beta}}{\partial q_{\gamma}} - \frac{\partial D_{\gamma\alpha}}{\partial q_{\beta}} \right) \dot{q}_{\alpha}\end{aligned}\tag{5.21}$$

So by symmetry of the Inertia matrix, we conclude that $\eta_{\gamma\beta} = -\eta_{\beta\gamma}$ and the matrix $\boldsymbol{\eta}$ is skew-symmetric.

Now we can prove the main result of this section:

Theorem 2. *The system given by equations (3.50) is passive with respect to the $\boldsymbol{\tau}$ - $\dot{\mathbf{q}}$ s input-output map.*

Consider the Hamiltonian (equal to the Lagrangian) in equation (3.47):

$$\mathcal{H} = \frac{1}{2} \dot{\mathbf{q}}^{\top} \mathbf{D} \dot{\mathbf{q}}.\tag{5.22}$$

Taking the derivative and integrating both sides from 0 to T yields:

$$\int_0^T \dot{\mathcal{H}} dt = \int_0^T \dot{\mathbf{q}}^{\top} \mathbf{D} \ddot{\mathbf{q}} + \frac{1}{2} \dot{\mathbf{q}}^{\top} \dot{\mathbf{D}} \dot{\mathbf{q}} dt\tag{5.23}$$

$$\mathcal{H}(T) - \mathcal{H}(0) = \int_0^T \dot{\mathbf{q}}^{\top} \left(\mathbf{D} \ddot{\mathbf{q}} + \frac{1}{2} \dot{\mathbf{D}} \dot{\mathbf{q}} \right) dt\tag{5.24}$$

Using the EOM (3.50) yields:

$$\begin{aligned}\mathcal{H}(T) - \mathcal{H}(0) &= \int_0^T \dot{\mathbf{q}}^{\top} \left(\mathbf{Q} + \frac{1}{2} \dot{\mathbf{D}} \dot{\mathbf{q}} - \mathbf{C} \dot{\mathbf{q}} \right) dt \\ &= \int_0^T \dot{\mathbf{q}}^{\top} \mathbf{Q} dt + \int_0^T \frac{1}{2} \dot{\mathbf{q}}^{\top} \left(\dot{\mathbf{D}} - 2\mathbf{C} \right) \dot{\mathbf{q}} dt \\ &= \int_0^T \dot{\mathbf{q}}^{\top} \mathbf{Q} dt + \int_0^T \frac{1}{2} \dot{\mathbf{q}}^{\top} \boldsymbol{\eta} \dot{\mathbf{q}} dt\end{aligned}\tag{5.25}$$

By Lemma 3, the matrix $\boldsymbol{\eta}$ is skew-symmetric. Thus, we have: $\mathbf{x}^{\top} \boldsymbol{\eta} \mathbf{x} = 0$, $\forall \mathbf{x} \in \mathbb{R}^n$ and equation 5.25 becomes:

$$\mathcal{H}(T) - \mathcal{H}(0) = \int_0^T \dot{\mathbf{q}}^{\top} \mathbf{Q} dt,\tag{5.26}$$

and:

$$\begin{aligned}
\mathcal{H}(T) - \mathcal{H}(0) &= \int_0^T \begin{bmatrix} \dot{\mathbf{q}}_b^\top & \dot{\mathbf{q}}_s^\top \end{bmatrix} \begin{bmatrix} \mathbf{0}_{3 \times 1} \\ \boldsymbol{\tau} \end{bmatrix} dt \\
&= \int_0^T \dot{\mathbf{q}}_s^\top \boldsymbol{\tau} dt \\
&= \langle \dot{\mathbf{q}}_s, \boldsymbol{\tau} \rangle_T
\end{aligned} \tag{5.27}$$

Thus, since the energy of the system is a positive scalar:

$$-\mathcal{H}(0) = \beta \leq \langle \dot{\mathbf{q}}_s, \boldsymbol{\tau} \rangle_T \tag{5.28}$$

which achieves the proof.

This section showed that the system, within the assumptions of the inviscid simulator, is passive. The passivity theorem may thus be applied to design a robust inner-loop controller for the eel, using a VSP controller. The next section thus focuses on the design of a VSP controller.

5.3 Optimal Design of a Robust VSP controller

Lemma 2 guarantees the VSP-ness of an LTI system if it is SPR. We thus seek a SPR LTI controller with state-space matrices $\mathbf{A}_c$, $\mathbf{B}_c$, $\mathbf{C}_c$ and $\mathbf{D}_c$ to be determined in an optimal way.

5.3.1 Methodology

There are many ways to design a SPR LTI system [73]. The optimal model-based approach proposed here inspires from a LQR-synthesis combined with the use of KYP lemma, and has been introduced by McLaren and Slater for the control of large space structures in [74]. The novelty proposed here is the extension of the method to underactuated systems (whose number of generalized coordinates (states) is not equal to the number of control inputs). Indeed, the LQR approach supposes that all states be controllable directly, which is obviously not the case here as no control action can be applied to the body-related variables. An adaptation of the method is thus proposed, involving linearization of sub-parts of the overall dynamics. Then, Riccati equation is used to synthesize the output matrix $\mathbf{C}_c$, the state matrix $\mathbf{A}_c$ is obtained as a consequence of the state feedback law in a LQR fashion, whereas KYP lemma is invoked for the design of the control matrix $\mathbf{B}_c$. Finally the feed-forward matrix $\mathbf{D}_c$ is chosen as a very small positive gain to avoid amplifying high frequency noise and make the system VSP. The chosen approach is thus dual in the sense that it combines the robustness of SPR design with the optimality of LQR. Note that the present method is but one approach; in practice, the design flexibility has virtually no limits as long as the controller is made SPR through KYP lemma. The LQR optimality is not a requirement, and other techniques may be used.

The design methodology is now explained in details.

Let us first consider the plant nonlinear dynamics, given by equations (3.50) and recalled here:

$$\mathbf{D}(t)\ddot{\mathbf{q}}(t) + \mathbf{C}(t)\dot{\mathbf{q}}(t) = \mathbf{H}\boldsymbol{\tau}(t). \tag{5.29}$$

with:

$$\mathbf{H} = \begin{bmatrix} \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times N-1} \\ \mathbf{0}_{N-1 \times 3} & \mathbf{I}_{N-1 \times N-1} \end{bmatrix} \quad (5.30)$$

In the context of LQR design for nonlinear systems, it is first desired to linearize the plant about an equilibrium point in order to build a linear state observer of the plant in state-space form.

In their paper [74], McLaren and Slater linearize their plant about an equilibrium point, which makes sense because the state of the system can be steered directly toward the equilibrium point (around which the linearization is valid). However, this is no longer the case here: the so-called body-related variables $\mathbf{q}_b$ are not directly controllable in torque, by contrast to the shape variables $\mathbf{q}_s$. Concretely, it means that linearizing the plant about an equilibrium point does not really makes sense for the whole system in the context of the inner-loop control where the body variables cannot be steered to a desired position without the use of an outer-loop controller. Instead, the linearization should be performed on a sub-part of the whole system, namely on the sub-system describing the dynamics of the shape variables – which can be steered toward the equilibrium point in inner loop.

Introducing the following partitioning between the shape and body parts of the Inertia and Coriolis matrices:

$$\mathbf{D} = \left[\begin{array}{c|c} \mathbf{D}_{bb} & \mathbf{D}_{bs} \\ \hline \mathbf{D}_{sb} & \mathbf{D}_{ss} \end{array} \right] \quad (5.31)$$

$$\mathbf{C} = \left[\begin{array}{c|c} \mathbf{C}_{bb} & \mathbf{C}_{bs} \\ \hline \mathbf{C}_{sb} & \mathbf{C}_{ss} \end{array} \right] \quad (5.32)$$

with $\mathbf{D}_{bb}, \mathbf{C}_{bb} \in \mathbb{R}^{3 \times 3}$, $\mathbf{D}_{bs}, \mathbf{C}_{bs} \in \mathbb{R}^{3 \times N-1}$, $\mathbf{D}_{sb}, \mathbf{C}_{sb} \in \mathbb{R}^{N-1 \times 3}$ and $\mathbf{D}_{ss}, \mathbf{C}_{ss} \in \mathbb{R}^{N-1 \times N-1}$, the sub-system of the shape dynamics is thus given by:

$$\mathbf{D}_{sb}\ddot{\mathbf{q}}_b(t) + \mathbf{D}_{ss}\ddot{\mathbf{q}}_s(t) + \mathbf{C}_{sb}(t)\dot{\mathbf{q}}_b(t) + \mathbf{C}_{ss}(t)\dot{\mathbf{q}}_s(t) = \boldsymbol{\tau}, \quad (5.33)$$

where the matrices $\mathbf{D}_{ss}$ and $\mathbf{C}_{ss}$ are shape-related sub-blocks of respectively the full $\mathbf{D}$ and $\mathbf{C}$ matrices, and $\mathbf{D}_{sb}$ and $\mathbf{C}_{sb}$ are body-related sub-blocks.

Nevertheless, the shape-dynamics subsystem above is still coupled to the body variables via the $\mathbf{D}_{sb}$ and $\mathbf{C}_{sb}$ matrices, and still cannot be linearized about the shape configuration only. In order to get rid of the parasitic body dynamics in (5.33), a feedback linearization prewrap $\boldsymbol{\tau}^l$ such that $\boldsymbol{\tau} = \boldsymbol{\tau}_* + \boldsymbol{\tau}^l = \boldsymbol{\tau}_* + \hat{\mathbf{D}}_{sb}\ddot{\mathbf{q}}_b(t) + \hat{\mathbf{C}}_{sb}(t)\dot{\mathbf{q}}_b(t)$, should strictly speaking be applied as follows:

$$\cancel{\mathbf{D}_{sb}\ddot{\mathbf{q}}_b(t)} + \mathbf{D}_{ss}\ddot{\mathbf{q}}_s(t) + \cancel{\mathbf{C}_{sb}(t)\dot{\mathbf{q}}_b(t)} + \mathbf{C}_{ss}(t)\dot{\mathbf{q}}_s(t) = \boldsymbol{\tau}_* + \hat{\mathbf{D}}_{sb}\ddot{\mathbf{q}}_b(t) + \hat{\mathbf{C}}_{sb}(t)\dot{\mathbf{q}}_b(t), \quad (5.34)$$

where the $\hat{\cdot}$ notation stands for estimated matrices evaluated at the current configuration.

The cancelled shape dynamics thus becomes²:

$$\mathbf{D}_{ss}\ddot{\mathbf{q}}_s(t) + \mathbf{C}_{ss}(t)\dot{\mathbf{q}}_s(t) = \boldsymbol{\tau}_*, \quad (5.35)$$

and a proper linearization would thus be performed about an equilibrium shape configuration (obviously the point $\dot{\mathbf{q}}_s = \mathbf{0}$ as the joints oscillate around this position in case of forward motion). In practice, however, the feedback linearization pre-wrap alters significantly the performances and may be prohibitive if the model of the plant does not fit well with reality. So, in the present approach, the mathematical rigour is sacrificed at the benefit of improving the performances and the following is done: the linearization is performed on the decoupled shape-related sub-plant in equation (5.35), but the feedback linearization step is not added, meaning that the effective shape-related sub-plant is still (5.33) and the control input is just $\boldsymbol{\tau} = \boldsymbol{\tau}_*$ without the cancelling terms. In other words, the body coupling in the shape dynamics is ignored and the shape decoupled sub-plant (5.35) is regarded as a good proxy for the overall shape dynamics (5.33), which is a strong and arguable assumption.

The motivation for ignoring the body-coupling in the shape sub-block dynamics arises from the fact that the inner-loop controller we are designing in this section acts only on the shape variables and can thus be rendered "blind" to the body variable dynamics. Moreover, it is most likely that the dynamics of the shape variables evolves on a smaller time scale than the dynamics of the body variables: typically, variations in the pose of the eel are occurring at the rate of the outer-loop controller, whose update rate is significantly lower than the update rate of the inner-loop controller (driving the change of the joint angles). Finally, let us note that even in the worst case in which the model does not match with the process, only the optimal benefit of the LQR approach is lost, without altering the robustness properties guaranteed by the SPR-ness/VSP-ness through passivity theorem³. In practice it will be shown latter that despite this strong assumption is made, the optimal properties inherent to the LQR synthesis are conserved.

So, in order to linearize the decoupled sub-block in (5.35), the following state-space form realization is considered:

$$\dot{\mathbf{x}}(t) = -\mathbf{D}_{ss}^{-1}(t)\mathbf{C}_{ss}(t)\mathbf{x}(t) + \mathbf{D}_{ss}^{-1}(t)\boldsymbol{\tau}_*, \quad (5.36)$$

$$\mathbf{y} = \mathbf{x}, \quad (5.37)$$

with the state $\mathbf{x} = \dot{\mathbf{q}}_s$ and output⁴ $\mathbf{y} = \dot{\mathbf{q}}_s$. The linearization of the shape dynamics is now performed about the equilibrium point $\dot{\mathbf{q}}_s = \mathbf{0}$ and yields the (square) matrices $\mathbf{A}_p$ and $\mathbf{B}_p$.

²It may be shown straightforwardly by adaptation of the demonstration of theorem 2 that this new feedback-linearization-prewrapped system has also a passive I-O map.

³There is indeed a large flexibility in the choice of the state matrices of the LTI controller: as long as the controller is SPR (and then made VSP), the passivity theorem still applies and the stability properties of the closed loop are guaranteed, no matter how suboptimal the matrices are or no matter if they fit the plant or not.

⁴The choice of the shape variables as an output to the system is motivated by the passivity theorem: the robust feedback interconnection has to be achieved using the input-output map with respect to which the system is passive.

As in the LQR framework, let us now consider a Luenberger observer of the plant, whose dynamical equations are given by:

$$\hat{\mathbf{x}}(t) = \mathbf{A}_p \hat{\mathbf{x}}(t) + \mathbf{B}_p \boldsymbol{\tau}(t) + \mathbf{B}_c(\mathbf{y}(t) - \hat{\mathbf{y}}(t)) \quad (5.38)$$

with $\hat{\mathbf{y}}(t) = \hat{\mathbf{x}}(t)$. The $\hat{\cdot}$ notation accounts for the estimated nature of the state and output (the estimates $\hat{\mathbf{x}}$ and $\hat{\mathbf{y}}$) by contrast to the real state and output ($\mathbf{x}$ and $\mathbf{y}$). The correction term $\mathbf{B}_c(\mathbf{y}(t) - \hat{\mathbf{y}}(t))$ is used to correct the dynamics of the estimated state as a function of the error between the measured output and the reconstructed output. Also, $\mathbf{B}_c$ is the Luenberger matrix, and $\mathbf{y} = \dot{\mathbf{q}}_s$ as defined previously. This state observer aims at mimicking and dynamically estimating the behavior of the nonlinear plant via the linearized version.

A state feedback law of the form $\boldsymbol{\tau} = -\mathbf{C}_c \hat{\mathbf{x}}$ is now fed back to the plant (and thus to the observer) in a LQR fashion, achieving the interconnection of the Luenberger observer with the plant. The dynamics of the Luenberger observer + state feedback controller thus becomes:

$$\hat{\mathbf{x}}(t) = (\mathbf{A}_p - \mathbf{B}_p \mathbf{C}_c - \mathbf{B}_c) \hat{\mathbf{x}}(t) + \mathbf{B}_c \mathbf{y}(t) \quad (5.39)$$

$$\boldsymbol{\xi}(t) = \mathbf{C}_c \hat{\mathbf{x}}(t) \quad (5.40)$$

or, introducing $\mathbf{A}_c = \mathbf{A}_p - \mathbf{B}_p \mathbf{C}_c - \mathbf{B}_c$:

$$\hat{\mathbf{x}}(t) = \mathbf{A}_c \hat{\mathbf{x}}(t) + \mathbf{B}_c \mathbf{y}(t) \quad (5.41)$$

$$\boldsymbol{\xi}(t) = \mathbf{C}_c \hat{\mathbf{x}}(t) \quad (5.42)$$

Note that:

- the input of the state observer is the output of the plant: $\mathbf{y} = \dot{\mathbf{q}}_s$.
- the output of the state observer is the (opposite of the) input of the plant: $\boldsymbol{\xi} = -\boldsymbol{\tau}$.

Thus, the system formed by the Luenberger observer with the state-feedback gain is the desired controller, and the negative state feedback law achieves the negative feedback interconnection suggested by the passivity theorem. Now it remains to choose the matrices of the controller:

Design of $\mathbf{C}_c$. The choice for the matrix $\mathbf{C}_c$ follows directly from the classical LQR theory: $\mathbf{C}_c$ is the gain matrix of the state-feedback law, chosen as to solve a quadratic minimization problem. The problem consists in dynamically choosing the input to the nonlinear plant (and thus the matrix $\mathbf{C}_c$) so that the following quadratic performance criterion is minimized:

$$\min_{\mathbf{u}} J = \int_0^T \mathbf{x}^\top \mathbf{Q}_1 \mathbf{x} + \mathbf{u}^\top \mathbf{R} \mathbf{u} dt, \quad (5.43)$$

where $\mathbf{Q}_1 \geq 0$ and $\mathbf{R} > 0$ are tunable penalty matrices allowing a trade-off between actuator consumption and convergence speed of the system. For example, if one take $\mathbf{Q}_1 = q\mathbf{I}$ and $\mathbf{R} = \mathbf{I}$ (where $\mathbf{I}$ is the identity with proper dimensions), decreasing q would penalize the input and thus yield more energy efficient responses, at the expense of convergence

performances.

When the horizon is infinite (i.e $T = \infty$), it is well known that the solution to the above problem consists in choosing an input of the form $\boldsymbol{\tau} = -\mathbf{C}_c \hat{\mathbf{x}}$ (already enforced previously) where the output matrix is given by $\mathbf{C}_c = \mathbf{R}^{-1} \mathbf{B}_p^\top \mathbf{X}$, with $\mathbf{X}$ solution of the following Algebraic Riccati Equation (ARE):

$$\mathbf{X} \mathbf{A}_p + \mathbf{A}_p^\top \mathbf{X} - \mathbf{X} \mathbf{B}_p \mathbf{R}^{-1} \mathbf{B}_p^\top \mathbf{X} + \mathbf{Q}_1 = \mathbf{0}. \quad (5.44)$$

The latter equation can be solved numerically.

Design of $\mathbf{B}_c$. The matrix $\mathbf{B}_c$ is designed to make the controller SPR. Invoking KYP lemma, it can be chosen as:

$$\mathbf{B}_c = \mathbf{P}^{-1} \mathbf{C}_c^\top, \quad (5.45)$$

where $\mathbf{P} = \mathbf{P}^\top > 0$ is solution of the Lyapunov equation:

$$\mathbf{P} \mathbf{A}_c + \mathbf{A}_c^\top \mathbf{P} = -\mathbf{Q}_2, \quad (5.46)$$

where $\mathbf{Q}_2 = \mathbf{Q}_2^\top > 0$. Introducing $\mathbf{A}_c = \mathbf{A}_p - \mathbf{B}_p \mathbf{C}_c - \mathbf{B}_c = \mathbf{A}_* - \mathbf{B}_c$ in equation 5.46 yields:

$$\mathbf{P} \mathbf{A}_* + \mathbf{A}_*^\top \mathbf{P} = -(\mathbf{Q}_2 - \mathbf{C}_c^\top - \mathbf{C}_c) = -\mathbf{Q}_3, \quad (5.47)$$

The matrix $\mathbf{Q}_3$ should be symmetric definite positive for the solution $\mathbf{P}$ to be unique. In practice, this can be achieved for $\mathbf{Q}_2$ sufficiently high.

Design of $\mathbf{A}_c$. The matrix $\mathbf{A}_c$ follows directly from its definition: $\mathbf{A}_c = \mathbf{A}_p - \mathbf{B}_p \mathbf{C}_c - \mathbf{B}_c$, but the user has to check *a posteriori* that it is Hurwitz as nothing guarantees it. It is also possible to choose $\mathbf{A}_c = \mathbf{A}_p - \mathbf{B}_p \mathbf{C}_c$ in case the previous choice is not Hurwitz, as explained in [73].

Design of $\mathbf{D}_c$. Finally, in agreement with lemma 2, the controller is made VSP by adding a feed-forward matrix of the form:

$$\mathbf{D}_c = \delta \mathbf{I}. \quad (5.48)$$

with a small $\delta > 0$ in order to avoid amplifying high frequency noise.

Note that the design method presented here is but one approach to synthesize a SPR LTI. Other approaches are described in the literature [73]; some are using more sophisticated design procedures such as the $\mathcal{H}_\infty$ -synthesis.

5.3.2 Trajectory Following

As in the LQR framework, the controller above stabilizes the plant about the origin. In order to achieve trajectory tracking in joint space, the error between the desired and effective trajectory has to be fed-back to the plant input. However, care must be taken as to make sure that the addition of a feedback loop to the plant will not destroy the passive mapping guaranteed by theorem 2. The following theorem states that proportional action does not destroy passivity:

Theorem 3 (Conservation of Passivity). *A proportional control prewrap of the form: $\tau^p = -k_p \Delta \mathbf{q}_s = -k_p(\mathbf{q}_s - \mathbf{q}_s^*)$, $k_p > 0$, applied to the system (5.29), conserves passivity with respect to the $\tau - \dot{\mathbf{q}}_s$ input-output map⁵.*

Consider the energy-like function:

$$\mathcal{V} = \frac{1}{2} \dot{\mathbf{q}}^\top \mathbf{D} \dot{\mathbf{q}} + \frac{1}{2} k_p \Delta \mathbf{q}_s^\top \Delta \mathbf{q}_s. \quad (5.49)$$

Taking the derivative and integrating both sides from 0 to T yields:

$$\int_0^T \dot{\mathcal{V}} dt = \int_0^T \dot{\mathbf{q}}^\top \mathbf{D} \ddot{\mathbf{q}} + \frac{1}{2} \dot{\mathbf{q}}^\top \dot{\mathbf{D}} \dot{\mathbf{q}} + k_p \dot{\mathbf{q}}_s^\top \Delta \mathbf{q}_s dt, \quad (5.50)$$

$$\mathcal{V}(T) - \mathcal{V}(0) = \int_0^T \dot{\mathbf{q}}^\top \left(\mathbf{D} \ddot{\mathbf{q}} + \frac{1}{2} \dot{\mathbf{D}} \dot{\mathbf{q}} \right) + k_p \dot{\mathbf{q}}_s^\top \Delta \mathbf{q}_s dt. \quad (5.51)$$

Using the EOM (5.29) yields:

$$\begin{aligned} \mathcal{V}(T) - \mathcal{V}(0) &= \int_0^T \dot{\mathbf{q}}^\top \left(\mathbf{Q} + \frac{1}{2} \dot{\mathbf{D}} \dot{\mathbf{q}} - \mathbf{C} \dot{\mathbf{q}} \right) + k_p \dot{\mathbf{q}}_s^\top \Delta \mathbf{q}_s dt, \\ &= \int_0^T \dot{\mathbf{q}}^\top \mathbf{Q} dt + \frac{1}{2} \dot{\mathbf{q}}^\top \boldsymbol{\eta} \dot{\mathbf{q}} + k_p \dot{\mathbf{q}}_s^\top \Delta \mathbf{q}_s dt, \\ &= \int_0^T \dot{\mathbf{q}}_s^\top (\boldsymbol{\tau} + \boldsymbol{\tau}^p) + k_p \dot{\mathbf{q}}_s^\top \Delta \mathbf{q}_s dt, \end{aligned} \quad (5.52)$$

where lemma 3 was invoked (skew-symmetry of $\boldsymbol{\eta}$). Now introducing the prewrap yields:

$$\begin{aligned} \mathcal{V}(T) - \mathcal{V}(0) &= \int_0^T \dot{\mathbf{q}}_s^\top (\boldsymbol{\tau} - k_p \Delta \mathbf{q}_s) + k_p \dot{\mathbf{q}}_s^\top \Delta \mathbf{q}_s dt, \\ &= \int_0^T \dot{\mathbf{q}}_s^\top \boldsymbol{\tau} dt, \end{aligned} \quad (5.53)$$

and thus, since the energy-like function chosen is a positive scalar:

$$-\mathcal{V}(0) = \beta \leq \langle \dot{\mathbf{q}}_s, \boldsymbol{\tau} \rangle_T, \quad (5.54)$$

which achieves the proof.

5.3.3 Feedback Interconnection and Stability Discussion

The negative feedback interconnection of the passive system discussed in section 5.2 with the VSP controller discussed in section 5.3 is shown in Figure 5.2. The obtained system is perturbed by external inputs $\mathbf{d}$ (disturbances) and $\mathbf{n}$ (noise).

The passivity theorem states that the closed loop is $\mathbb{L}_2$ -stable, or more informally robust to any bounded external input ("finite energy in yields finite energy out"). We will however show an even stronger result, related to Lyapunov stability theory, and valid when the closed loop is not submitted to any external input (autonomous system). The classical LaSalle's invariant set theorem in Lyapunov stability theory will be used in the demonstration of this result; the reader not aware of these results is referred to Appendix C for a statement of LaSalle's theorem and a reminder on Lyapunov stability concepts.

We are now in position to demonstrate the following theorem, characterizing the stability of the closed loop:

⁵The asterisk denotes a desired joint angle issued by a guidance controller (tracking problem).

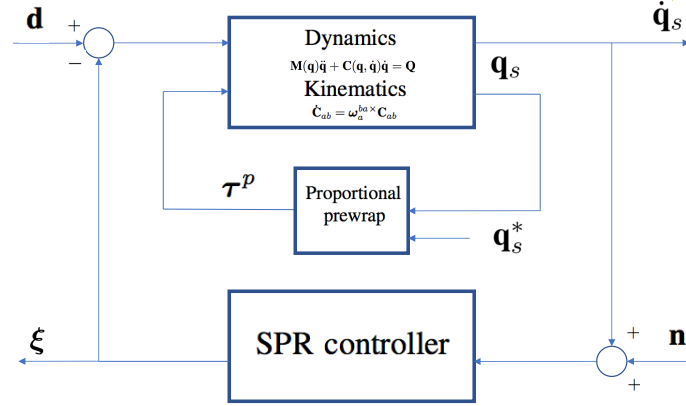


Figure 5.2: Feedback interconnection of a passive system with a linear SPR controller. Passivity theorem states that the closed loop is input-output stable.

Theorem 4 (Stability of the Closed Loop). *Let the nonlinear plant be described by (5.29):*

$$\mathbf{D}(t)\ddot{\mathbf{q}}(t) + \mathbf{C}(t)\dot{\mathbf{q}}(t) = \mathbf{H}\boldsymbol{\tau}(t), \quad (5.55)$$

and the VSP controller realising the negative feedback interconnection of Figure 5.2:

$$\hat{\mathbf{x}}(t) = \mathbf{A}_c \hat{\mathbf{x}}(t) + \mathbf{B}_c \dot{\mathbf{q}}_s(t) \quad (5.56)$$

$$\boldsymbol{\xi}(t) = \mathbf{C}_c \hat{\mathbf{x}}(t) + \mathbf{D}_c \dot{\mathbf{q}}_s(t) \quad (5.57)$$

$$\boldsymbol{\tau}(t) = -k_p \Delta \mathbf{q}_s(t) - \boldsymbol{\xi}(t) \quad (5.58)$$

where the triple $(\mathbf{A}_c, \mathbf{B}_c, \mathbf{C}_c)$ is SPR, $\mathbf{D}_c = \mathbf{D}_c^\top > 0$ and $k_p > 0$. Then, the (equilibrium) point $(\Delta \mathbf{q}_s, \dot{\mathbf{q}}, \hat{\mathbf{x}}) = (0, 0, 0)$ is asymptotically stable.

At first, note that the validity of $(\Delta \mathbf{q}_s, \dot{\mathbf{q}}, \hat{\mathbf{x}}) = (0, 0, 0)$ as an equilibrium point is checked trivially by introducing $\hat{\mathbf{x}} = 0$, $\ddot{\mathbf{q}} = 0$ and $\dot{\mathbf{q}} = 0$ in the equations (5.55) - (5.58).

Then, consider the energy-like function:

$$\mathcal{V} = \frac{1}{2} \dot{\mathbf{q}}^\top \mathbf{D} \dot{\mathbf{q}} + \frac{1}{2} k_p \Delta \mathbf{q}_s^\top \Delta \mathbf{q}_s + \frac{1}{2} \hat{\mathbf{x}}^\top \mathbf{P} \hat{\mathbf{x}}. \quad (5.59)$$

Differentiating with respect to time and expanding yields:

$$\begin{aligned}
\dot{\mathcal{V}} &= \frac{1}{2} \dot{\mathbf{q}}^T \dot{\mathbf{D}} \dot{\mathbf{q}} + \underbrace{\dot{\mathbf{q}}^T \mathbf{D} \ddot{\mathbf{q}}}_{-\dot{\mathbf{q}}^T \mathbf{C} \dot{\mathbf{q}} + \dot{\mathbf{q}}^T \mathbf{H} \boldsymbol{\tau}} + k_p \dot{\mathbf{q}}_s^T \Delta \mathbf{q}_s + \hat{\mathbf{x}}^T \mathbf{P} \underbrace{\hat{\mathbf{x}}}_{\mathbf{A}_c \hat{\mathbf{x}} + \mathbf{B}_c \dot{\mathbf{q}}}, \\
&= \frac{1}{2} \dot{\mathbf{q}}^T \dot{\mathbf{D}} \dot{\mathbf{q}} - \dot{\mathbf{q}}^T \mathbf{C} \dot{\mathbf{q}} + \underbrace{\dot{\mathbf{q}}^T \mathbf{H} \boldsymbol{\tau}}_{\dot{\mathbf{q}}_s^T} + k_p \dot{\mathbf{q}}_s^T \Delta \mathbf{q}_s + \overbrace{\hat{\mathbf{x}}^T \mathbf{P} \mathbf{A}_c \hat{\mathbf{x}} + \hat{\mathbf{x}}^T \underbrace{\mathbf{P} \mathbf{B}_c}_{\mathbf{C}_c^T} \dot{\mathbf{q}}_s}^{\text{KYP lemma}}, \\
&= \frac{1}{2} \underbrace{\dot{\mathbf{q}}^T (\dot{\mathbf{D}} - 2\mathbf{C}) \dot{\mathbf{q}}}_{0, \text{ skew-symmetric}} + \dot{\mathbf{q}}_s^T (-k_p \Delta \mathbf{q}_s - \boldsymbol{\xi}) + k_p \dot{\mathbf{q}}_s^T \Delta \mathbf{q}_s - \frac{1}{2} \hat{\mathbf{x}}^T \mathbf{Q}_2 \hat{\mathbf{x}} + \underbrace{(\mathbf{C}_c \hat{\mathbf{x}})^T}_{\boldsymbol{\xi} - \mathbf{D}_c \dot{\mathbf{q}}_s} \dot{\mathbf{q}}_s, \\
&= -\dot{\mathbf{q}}_s^T \boldsymbol{\xi} - \frac{1}{2} \hat{\mathbf{x}}^T \mathbf{Q}_2 \hat{\mathbf{x}} + \underbrace{\boldsymbol{\xi}^T \dot{\mathbf{q}}_s}_{\dot{\mathbf{q}}_s^T \boldsymbol{\xi}} - \dot{\mathbf{q}}_s^T \mathbf{D}_c^T \dot{\mathbf{q}}_s, \\
&= -\frac{1}{2} \hat{\mathbf{x}}^T \mathbf{Q}_2 \hat{\mathbf{x}} - \dot{\mathbf{q}}_s^T \mathbf{D}_c^T \dot{\mathbf{q}}_s, \\
&= -\frac{1}{2} \mathbf{z}^T \text{diag}(\mathbf{Q}_2, \mathbf{D}_c, \mathbf{0}) \mathbf{z}, \\
&\leq 0
\end{aligned} \tag{5.60}$$

where $\mathbf{z}$ is obtained by stacking $\hat{\mathbf{x}}$, $\dot{\mathbf{q}}_s$ and $\Delta \mathbf{q}_s$ together. The system is thus Lyapunov stable, but this result does not allow to conclude to asymptotic stability yet, since the matrix $\text{diag}(\mathbf{Q}_2, \mathbf{D}_c, \mathbf{0})$ is positive semi-definite.

Let us try with LaSalle's invariant set theorem (stated in Appendix C). At first, note:

$$\int_0^T \dot{\mathcal{V}}(\Delta \mathbf{q}_s, \dot{\mathbf{q}}, \hat{\mathbf{x}}) dt \leq 0, \tag{5.61}$$

$$\mathcal{V}(\Delta \mathbf{q}_s(T), \dot{\mathbf{q}}(T), \hat{\mathbf{x}}(T)) - \mathcal{V}(\Delta \mathbf{q}_s(0), \dot{\mathbf{q}}(0), \hat{\mathbf{x}}(0)) \leq 0, \tag{5.62}$$

$$\mathcal{V}(\Delta \mathbf{q}_s(T), \dot{\mathbf{q}}(T), \hat{\mathbf{x}}(T)) \leq \mathcal{V}(\Delta \mathbf{q}_s(0), \dot{\mathbf{q}}(0), \hat{\mathbf{x}}(0)). \tag{5.63}$$

As such, the compact set Ω can be defined by:

$$\Omega = \{(\Delta \mathbf{q}_s, \dot{\mathbf{q}}, \hat{\mathbf{x}}) \in \mathbb{R}^n \mid \mathcal{V}(\Delta \mathbf{q}_s(T), \dot{\mathbf{q}}(T), \hat{\mathbf{x}}(T)) \leq \mathcal{V}(\Delta \mathbf{q}_s(0), \dot{\mathbf{q}}(0), \hat{\mathbf{x}}(0))\}. \tag{5.64}$$

Now consider the set:

$$E = \{(\Delta \mathbf{q}_s, \dot{\mathbf{q}}, \hat{\mathbf{x}}) \in \Omega \mid \dot{\mathcal{V}} = 0\}. \tag{5.65}$$

This set is populated by $\hat{\mathbf{x}} = 0$ and $\dot{\mathbf{q}}_s = 0$ since $\dot{\mathcal{V}} = 0 \Leftrightarrow -\frac{1}{2} \hat{\mathbf{x}}^T \mathbf{Q}_2 \hat{\mathbf{x}} - \dot{\mathbf{q}}_s^T \mathbf{D}_c^T \dot{\mathbf{q}}_s = 0$. Let us now build the largest invariant set in E .

The result $\dot{\mathbf{q}}_s = 0$ implies that $\ddot{\mathbf{q}}_s = 0$. Thus, from the EOM: $\boldsymbol{\tau} = 0 \Leftrightarrow 0 = -k_p \Delta \mathbf{q}_s - \boldsymbol{\xi}$. Replacing by $\boldsymbol{\xi} = \underbrace{\mathbf{C}_c \hat{\mathbf{x}}}_0 + \underbrace{\mathbf{D}_c \dot{\mathbf{q}}_s}_0 = 0$, the conclusion is $\Delta \mathbf{q}_s = 0$.

Thus, the largest invariant set in E is:

$$M = \{(\Delta \mathbf{q}_s, \dot{\mathbf{q}}, \hat{\mathbf{x}}) \in E \mid (\Delta \mathbf{q}_s(t), \dot{\mathbf{q}}(t), \hat{\mathbf{x}}(t)) = 0\}. \quad (5.66)$$

As a conclusion, from LaSalle's invariant set theorem, the trajectories of the closed-loop asymptotically approach M , i.e. $(\Delta \mathbf{q}_s(t), \dot{\mathbf{q}}(t), \hat{\mathbf{x}}(t)) \rightarrow (0, 0, 0)$ as $t \rightarrow \infty$.

Theorem 4 states that the proposed controller stabilizes asymptotically the system around a desired trajectory $(\Delta \mathbf{q}(t) \rightarrow 0)$, in the absence of disturbances⁶ (autonomous system). The passivity theorem then takes over in the presence of a wide envelope of disturbances (robustness via $\mathbb{L}_2$ -stability).

These theoretical results offer a comfortable margin in terms of stability and robustness. The next section deals with assessing the highlighted performances against simulation results.

5.4 Simulations

5.4.1 Methodology

The controller discussed in the previous section is synthesized for a $N = 3$ links eel-like robot, based on the nonlinear inviscid model developed in Chapter 3. The synthesis can be automated easily and sizes directly to any number N of links⁷. All state-space matrices are pre-computed offline, which prevents any computational burden. When the controller has to be implemented on digital systems – as is the case for the VPM-2D simulator – the continuous-time realization no longer holds. In practice, a discrete-time realization of this controller is obtained via either a Zero Order Hold (ZOH) or a Tustin approximation method – a ZOH discretization was used here, see Appendix A.

The following simulations will be performed (on the inviscid simulator, otherwise mentioned):

- Comparison of the performances of the SPR with a benchmark controller in the presence of noise / disturbances.
- Simulation of the SPR with various levels of noise / disturbances.
- Effect of the initial conditions on the trajectory tracking.
- Effect of actuator saturation.
- Effect of varying the tunable parameters of the controller (e.g. penalty matrices).
- Effect of unmodelled dynamics: simulation of a discrete-time implementation on VPM-2D.

For each simulation, the following parameters are used (referring to the notations of the previous Chapters): $L = 0.4\text{m}$, $b = 0.02\text{m}$, $c = 0.04\text{m}$, $\nu = 1\text{e-}4 \text{ m}^2/\text{s}$ (only for VPM-2D⁸),

⁶When large external inputs are applied on the closed loop through disturbances and noise, the system is no longer autonomous and the asymptotic property drops. However, the system remains Lyapunov stable as the time derivative of the energy-like function (Lyapunov function) is negative semi-definite, and the passivity theorem still enforces $\mathbb{L}_2$ -stability.

⁷And potentially to any plant dynamics as long as it has a passive map.

⁸The viscosity is increased in order to accelerate the simulations.

$$\rho_s = \rho = 1000 \text{ kg/m}^3.$$

Whenever robustness to external input has to be assessed, white noise is introduced either as a disturbance or as sensor noise. As a reminder, theoretically, white noise is an uncorrelated stochastic process with a constant Spectral Power Density (SPD) and infinite energy. In the following, the terms "noise level" or "noise power" refer to the SPD in [W], which is related to the signal magnitude squared.

As previously the results are first presented, then discussed in a separate section. Otherwise stated, when the term "trajectory" is used, it is intended "trajectory in joint space" (inner-loop control).

5.4.2 Results

At first, the closed loop obtained is tested in simulation against the introduction of disturbances: a 1 W SPD white noise is introduced as a disturbance (**d**, see Figure 5.2). The usual sinusoidal trajectory in joint space is generated, but this time it is not enforced at the joints as it was the case for the validation, but rather it is followed by the inner-loop controller. As a way of comparing the performances in terms of disturbance rejection, the experiment is carried out in the same conditions with the PD controller given by (2.2) (benchmark). The SPR is synthesized as explained in the previous section. Both controllers yield a perfect trajectory tracking in the absence of disturbances. Then, the disturbances are added, and the results are shown in Figure 5.3 for joint 2. The Figure shows that the system driven by SPR control still tracks the desired trajectory with no visible drift for a 1 W SPD white noise, whereas the behavior is more chaotic with PD control (mean errors of about 0.2 rad).

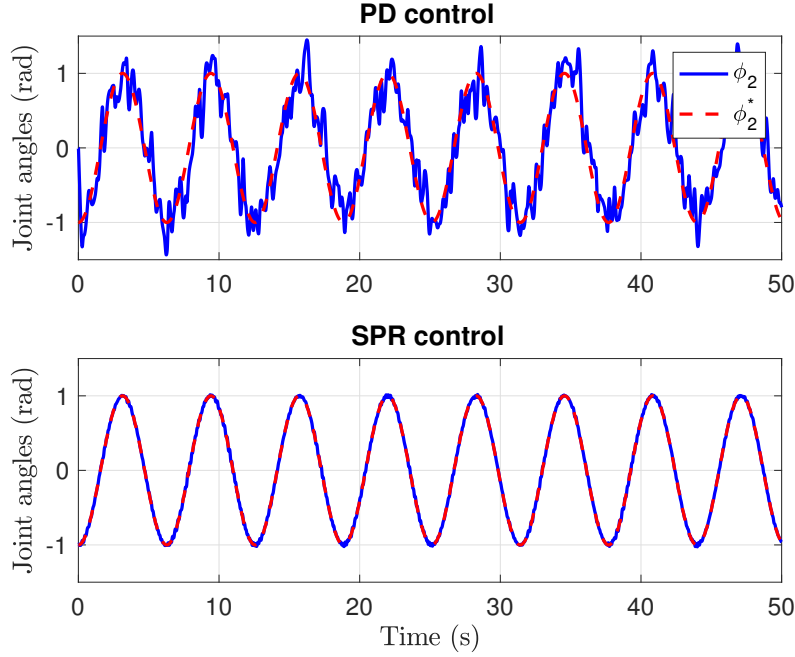


Figure 5.3: Comparison of PD and SPR controller in the event of the addition of disturbances to the system. Both controllers were initially fine tuned in the absence of external inputs.

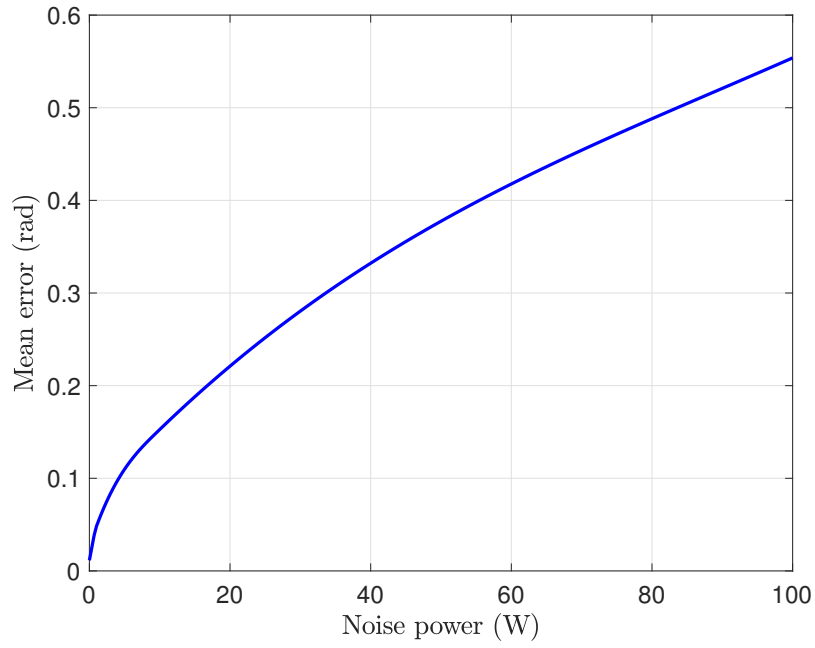


Figure 5.4: Evolution of the mean error between desired and achieved trajectory for different disturbance SPD.

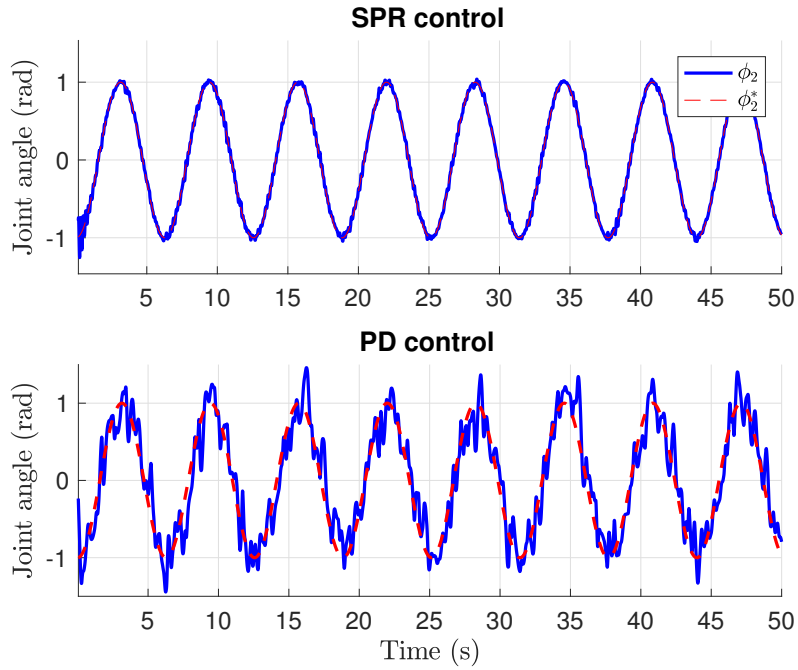


Figure 5.5: Comparison of PD and SPR controller in the event of the addition of noise to the system. Both controllers were initially fine tuned in the absence of external inputs.

The next set of simulations aims at assessing the limitations of the SPR controller in terms of disturbance rejection. Figure 5.4 is obtained by simulating the SPR-controlled plant at various SPD levels: for each power level of disturbance, the mean error between the reference signal and the achieved one is collected. The noise level is progressively

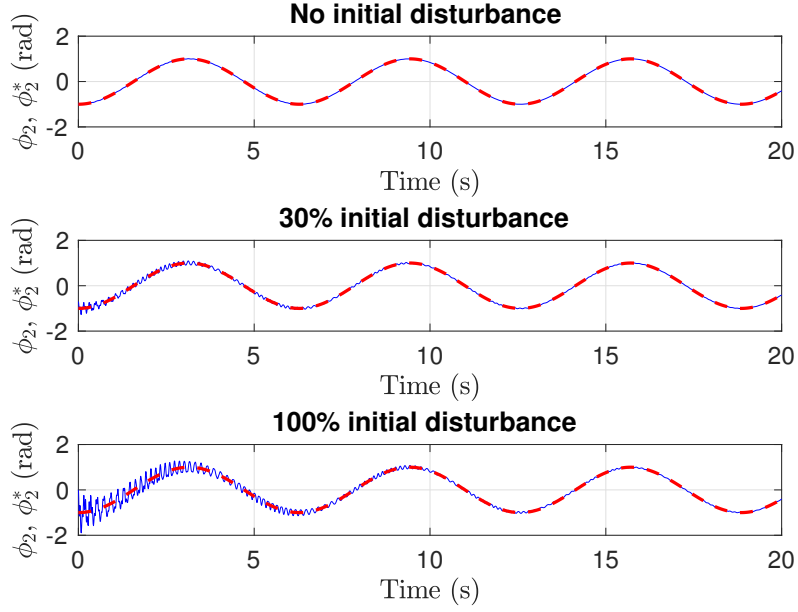


Figure 5.6: Effect of initial disturbances on the closed loop.

increased in each new simulation. For a SPD of about 20 W (which corresponds to a disturbance torque of about 5 N.m), the error reaches 0.2 rad – the same mean error than for the PD-controller in Figure 5.2.

The sensitivity of the two controllers to sensor noise is tested in simulation in the same fashion than previously: a 1 W SPD white noise is introduced in the system as a sensor noise ($\mathbf{n}$, see Figure 5.2). The initially fine tuned controllers are now perturbed by this noise level, as seen in Figure 5.5. This time, the sensitivity of the SPR controller to sensor noise is more important for the same SPD level as small drift oscillations are now visible.

Figure 5.6 highlights the sensitivity of the closed loop to initial conditions (in the absence of external inputs). Trajectory tracking in joint space is achieved for different initial conditions: when the system starts with a zero cross-track error, there are no discrepancies between the achieved and desired trajectories, but when the cross track error increases, ringing appears. In the second subplot, the system initial condition for joint 2 is $\phi_2(0) = -0.7$ while the desired initial position is $\phi_2^*(0) = -1$. This causes the system to oscillate slightly at start, before locking to the desired trajectory afterwards. These start up oscillations increase even more for the initial condition $\phi_2(0) = 0$, before being damped out afterwards.

The next simulations aim at evaluating the situations in which the passivity theorem may be violated.

At first, the effect of unmodelled dynamics is assessed via VPM-2D simulations, see Figures 5.8 and 5.9 . Note that the simulation was initiated with a discrepancy between the initial configuration and desired initial configuration to assess the effect of the initial conditions. Figure 5.8 compares the evolution of the desired and effective trajectories of the joint angles for both actuators: overall, the trajectory tracking is efficient. 5.9 focuses on

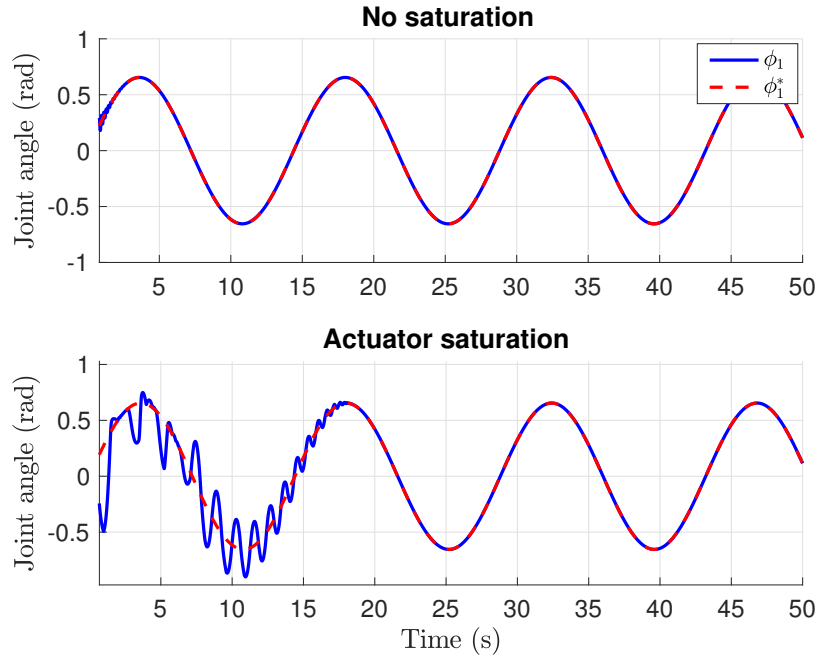


Figure 5.7: Effect of torque saturation on the closed loop.

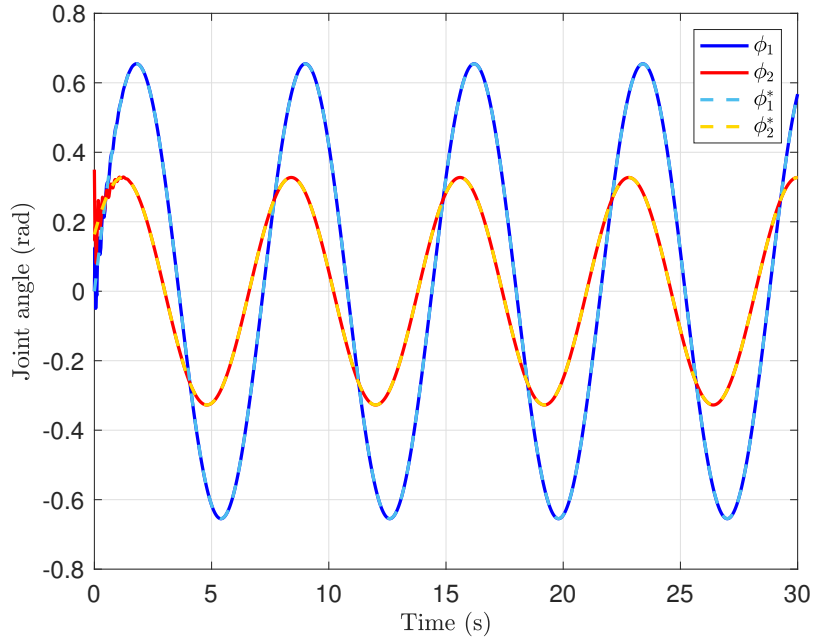


Figure 5.8: Effect of unmodelled dynamics (VPM-2D simulation) and initial disturbances on the closed loop.

the very first seconds of the simulation to show how the closed loop reacts at the beginning: small and quickly damped oscillations (ringing) occurs before the trajectories stabilize.

Then, the effect of actuator saturation is investigated in Figures 5.7 for the inviscid simulator and 5.10 for the VPM-2D solver. In the upper subplot of Figure 5.7, trajectory

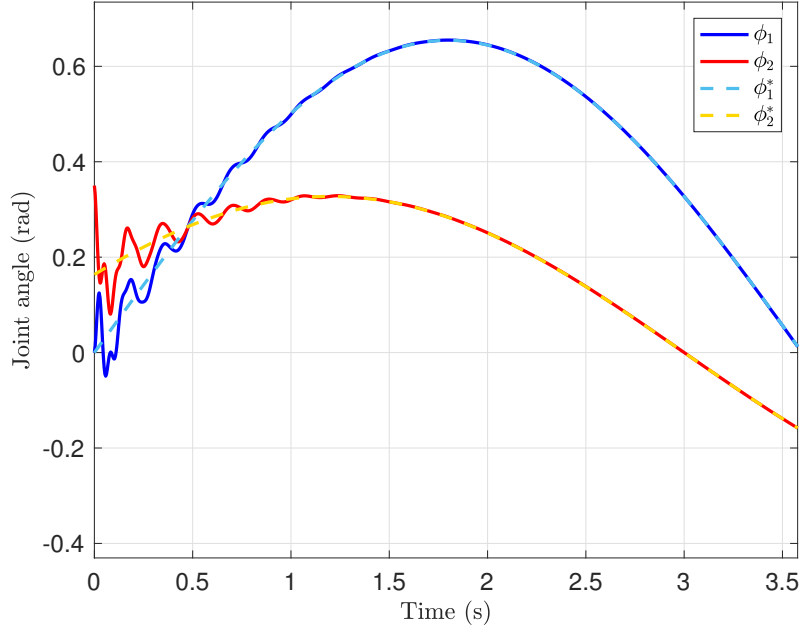


Figure 5.9: Zoom on Figure 5.8.

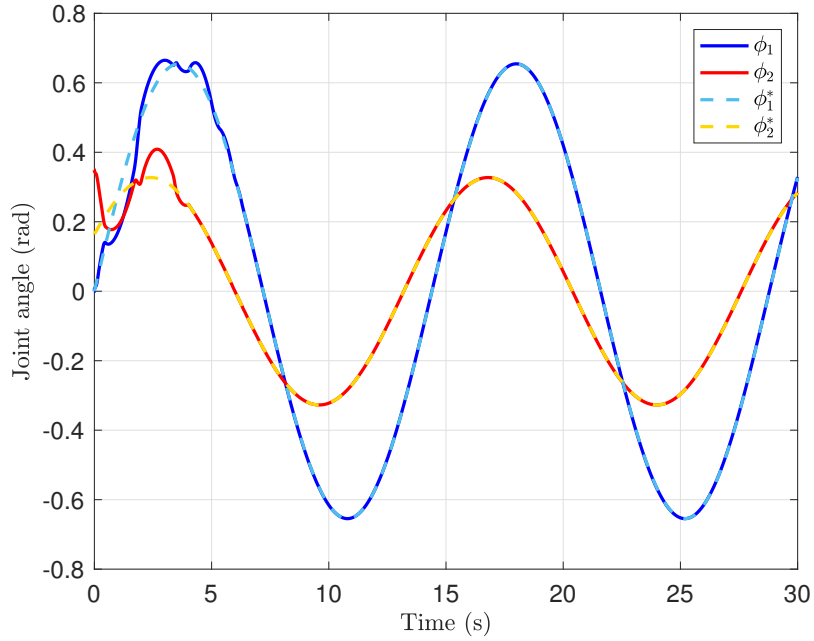


Figure 5.10: Effect of unmodelled dynamics (VPM-2D simulation) and actuator saturation on the closed loop.

tracking for joint 1 is shown without any saturation. The lower subplot shows the poor performances achieved in the event of actuator saturation: a maximum / minimum torque of ± 2 N.m is enforced (any commanded torque in excess would be capped to one of these thresholds), and the performances are momentarily affected, before the system recovers. The same effect is observed in Figure 5.10 where the simulation was performed with the

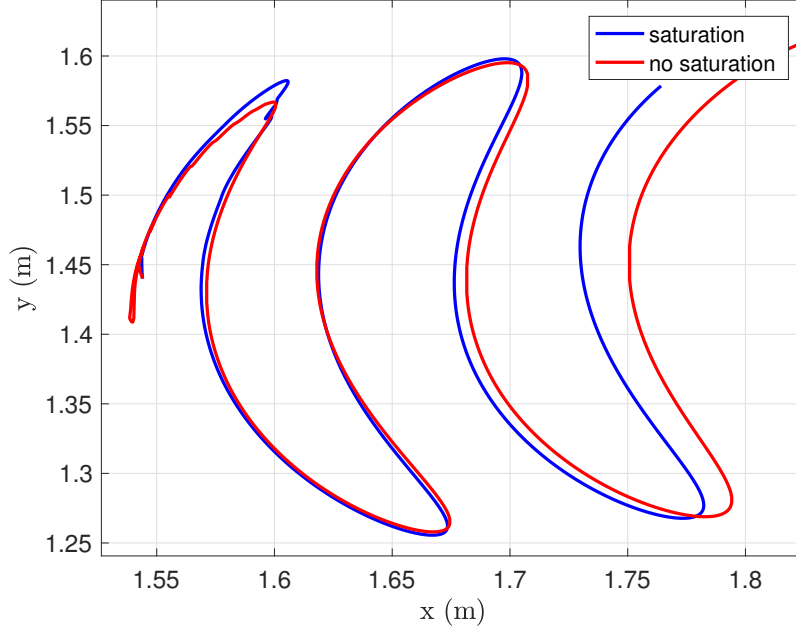


Figure 5.11: Effect of saturation in the x, y -path.

same saturation conditions (but with a different reference gait) on the VPM-2D simulator. A slighter perturbation occurs at the beginning, before the system locks on the desired trajectories in joint space.

Figure 5.11 shows the influence of the slight perturbation in joint space from actuator saturation in the x, y -path of the swimmer: the trajectory of the articulated body centroid is reported in the x, y -path for the case without saturation (smooth trajectory tracking in joint space) and with actuation (perturbation in joint space). A discrepancy of a few cm is observed between both paths, showing the sensitivity of the swimmer's body space trajectory to perturbations in shape space, and the importance to alleviate for the latter.

Finally the influence of the design parameters of the SPR is investigated. The most interesting parameters for which the designer has wide liberty in the design process are the $\mathbf{Q}_1$ and $\mathbf{R}$ matrices in the minimization criterion (5.43). Taking as suggested before: $\mathbf{Q}_1 = q\mathbf{I}$ and $\mathbf{R} = \mathbf{I}$, it is possible to adapt the trade-off existing between fast convergence and low-energy control. Two cases were simulated and reported in Figure 5.12: the above sub-plot stands for a good trajectory tracking with fast convergence ($q = 1000$), whereas the sub-plot beneath shows slower tracking performances, with a constant delay between the desired and effective trajectories ($q = 30$). Nevertheless, from Figure 5.13 comparing the torque evolution for the two cases, faster convergence comes at the expense of an aggressive input torque profile (blue curve) while slower convergence is achieved with a less demanding torque input (red curve).

5.4.3 Discussion

Figures 5.3 and 5.4 show the remarkable disturbance rejection capabilities of the SPR controller. In the first one, the SPR controller is shown to outperform the benchmark for a given SPD level: the PD-controlled closed loop suffers from important degradation of performances, whereas the SPR-controlled closed loop does not suffer from external inputs

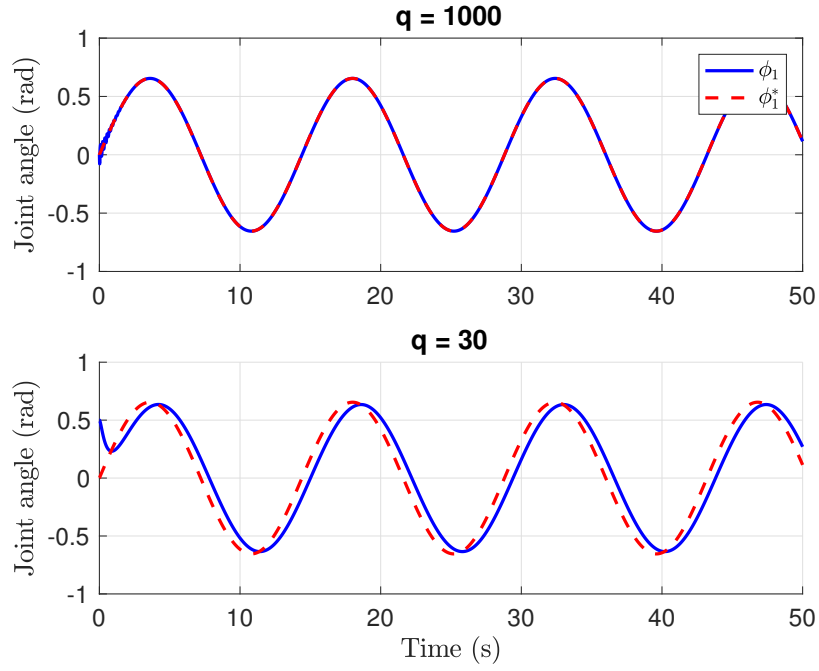


Figure 5.12: Effect of varying the penalty matrix on the performances of the closed loop.

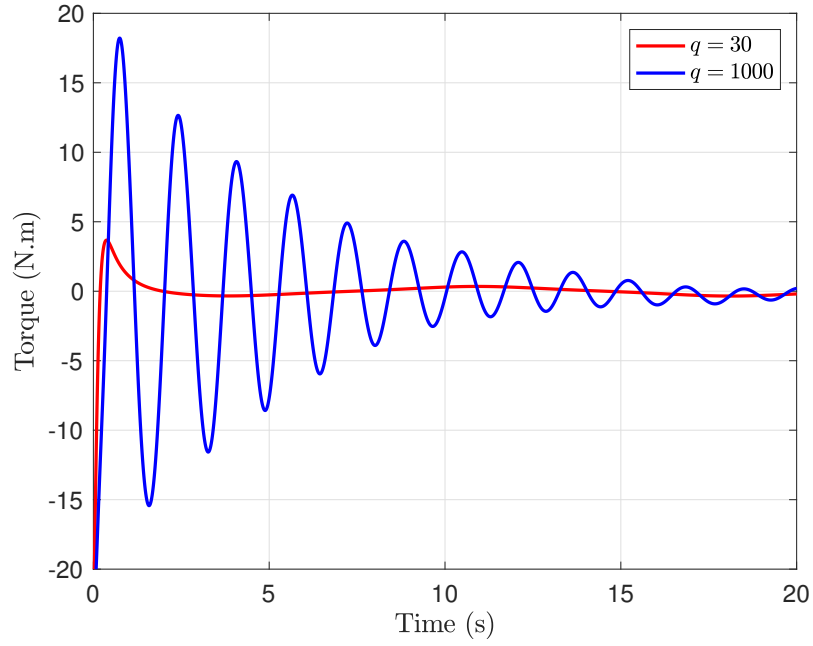


Figure 5.13: Effect of varying the penalty matrix on actuator torques.

at this power level. The second Figure shows, for the SPR-controlled plant, the evolution of the error between desired and achieved trajectories as a function of the disturbance level. For a given level of performance (a 0.2 rad mean error) there is half an order of magnitude between the disturbance levels rejected for the two controllers: the PD-controlled plant suffers from a mean error of 0.2 rad with a disturbance torque of 1 N.m whereas the SPR-controlled plant achieves the same performances with a disturbance of 5 N.m. We

conclude that the SPR-controlled plant is 5 times more robust to disturbances than the PD-controlled plant.

Figure 5.5 shows that the SPR controller is more sensitive to sensor noise (for a given SPD level) compared with the disturbance rejection performances in 5.3. However the proposed controller still overperforms PD control in terms of noise filtering.

The SPR control approach thus meets in practice the robustness properties guaranteed by the passivity theorem ($\mathbb{L}_2$ -stability). The improvement from the benchmark controller is significant. However, it should be noted that these performances were expected in the context of the inviscid simulator, for which the plant is passive. Passivity may be destroyed in a variety of cases (actuator saturation, unmodelled dynamics, etc.) and in these situations, the closed loop may lose the $\mathbb{L}_2$ -stability guaranteed by the passivity theorem. The next set of simulations aimed at verifying the performances of the closed loop in all these situations in which the passivity theorem does not apply.

As shown in Figure 5.7, actuator saturation significantly alters the performances at the beginning (when the demanded input is high) before the system stabilizes. This occurs because the input-output map guaranteeing the passivity of the plant is actually not valid anymore in the event of saturation as the admissible inputs are capped to the saturation threshold. When the demanded inputs are within the saturation limits, the passive input-output map is restored and the system converges toward the desired trajectory.

Then, the sensitivity to the initial conditions was highlighted by Figures 5.6, 5.8 and 5.9 for both the inviscid simulator and the VPM-2D solver. This is because the control command is too aggressive, and thus introduces overshoot and ringing. This is one of the main drawback of the presented controller, but it can be alleviated easily in practice by either slowly initiating the system so that the desired initial configuration is collocated with the actual initial configuration (for example, this was done in Chapter 2 with the gait whose amplitude was progressively increased at start) or by decreasing the penalty on the state from equation (5.43) (the $\mathbf{Q}_1$ matrix) so that the actuator authority is decreased and the command made smoother. The latter solution however comes at the cost of a slower, less reactive tracking. The trade-off between small actuator authority and efficient regulation was highlighted by Figures 5.12 and 5.13: increasing the penalty on the state yields a good but energy-intensive tracking, and conversely. In practice, tuning the penalty matrix $\mathbf{Q}_1$ offers to the designer the liberty of choosing between energy efficiency or performances.

It is even possible to imagine that this parameter would be adapted online, depending on the requirements: should the system perform aggressive and fast maneuvers, the penalty can be increased; should the system enter an energy-saving mode, the parameter can be reduced. This would however require the system to solve the Riccati equation in real time to perform a new SPR synthesis online. This could be conceived if sufficient time is allocated to change the SPR structure, for example if the system switches from an energy-consuming to a rest mode with no emergency. In a biological analogy, it would be e.g the eel resting after a long hunt. If however the change has to be applied within one-tenth of a second, in case for example a predator chases the eel in our biological analogy, then the approach is jeopardized due to relatively high computation costs involved with ARE solving⁹.

⁹There are however recent advances in applied mathematics providing analytic solution for the ARE equation under some assumptions (e.g [75]). An interesting extension of this work would thus be to implement a real-time SPR synthesis. Beyond the energy-focused motivation, it may also be attractive to synthesize the SPR dynamically for implementing a gain-scheduled controller, as discussed later.

Finally, the results from the VPM-2D simulations show good performances of the proposed controller. Despite the SPR was designed based on an inviscid model (simplified), the robustness (passivity) properties are conserved when replicating the controller on the simulator. A slighter influence from saturation and initial conditions are observed with VPM-2D, maybe because the observed ringing in the inviscid simulator is damped out by viscous forces in VPM-2D.

Overall the SPR controller is robust to unmodelled dynamics (viscous forces) and is thus well suited to control the simulator. Since the transition from the inviscid to the viscous simulator was successfully achieved we can, without being reckless, draw conclusions on the potential success of an implementation on the robotic platform. Unfortunately this hypothesis could be tested for the reasons raised in Chapter 2 (in particular because the Dynamixel are not torque-controllable and the platform is unstable). Future work would involve testing the SPR controller with proper actuators and a stable platform.

5.5 SPR control: perspectives and limitations

To sum this up in a nutshell, from the discussion above and from the theoretical properties established previously, the advantages and drawbacks of SPR control are discussed up below.

The controller proposed has many advantages:

- Scalability. The synthesis can be adapted to any link number or even any plant as long as it has a passive map.
- Quick and automated synthesis. An algorithm was written following the cooking recipe detailed in section 5.3, and the synthesis can be realized within seconds, solving the Lyapunov equation and ARE numerically with MATLAB.
- Automatic synthesis.
- Offline computation. The synthesis is done offline; all that remains to do online is the integration of the state space model, so that this controller is computationally inexpensive. It is still unclear yet if it would be possible to synthesize the controller in real time; this would mean solving the ARE and Lyapunov equations fast enough, or obtain analytic solutions. The implications would be tremendous for at least two reasons: on the one hand it would allow linearizing the plant for each current configuration, yielding a nonlinear gain-scheduled controller (more on this below), and on the other hand it would allow real-time changes in strategy regarding the design parameters (e.g switch from an energy intensive mode to a rest mode).
- Robustness. Through passivity theorem, there is a $\mathbb{L}_2$ -stability guarantee of the closed loop, as long as the passive map is not violated.
- Asymptotic stability (in the absence of external input). Through theorem 4, the asymptotic stability of the closed loop is guaranteed as long as the external inputs are negligible.
- Optimality. In the sense of a quadratic criterion inspired from the LQR paradigm, and based on a linearized model of the plant.

- Flexibility. The controller can be tuned to achieve either aggressive maneuvers or slower tracking. An easy interpretation of the tunable parameters is available.

The controller has the following drawbacks:

- Model-based. A reliable model of the plant is needed to make a relevant use of the LQR-like synthesis. Note that even if a poor model is available, or even if no model at all is available, the controller still achieve robust (but suboptimal) convergence as long as the SPR-ness/VSP-ness is kept.
- Passivity-based. The passivity theorem may be violated in the presence of non-idealities: actuator saturation, faults, delays, unmodelled dynamics...
- Sensitivity to initial conditions and actuator torque.
- Conceptual difficulty. The approach proposed is way more complex than a PD controller.
- Linearity. The controller is LTI and the linearization is only valid around an equilibrium position. The nonlinear dynamics is thus totally ignored. This could be alleviated via gain schedulling: various SPR controllers are designed at different operating points offline, and they are interpolated online as done in [73]. Moreover, contrary to classical gain-schedulling where the stability of the controller can only be assessed empirically by extensive simulations, this SPR-based approach relies on the I-O stability theory for $\mathbb{L}_2$ -stability guarantees: indeed, it is possible to show that a network of linear SPR system is VSP and apply the passivity theorem. Finally, new extensions could also involve the online synthesis of SPR controllers, aiming at forming a continuous gain-scheduled network.

5.6 Conclusion

This Chapter focused on the design of a passivity-based robust controller. The model-based approach combines a robust (in the $\mathbb{L}_2$ -stability sense) and optimal (in a LQR sense) frameworks via the synthesis of a SPR LTI controller made VSP via KYP lemma. Strong theoretical results allow to guarantee the robustness properties of the closed loop, as long as the passivity properties are preserved. Such theoretical results on the robustness of the system are important in modern control theory as more and more autonomous agents are going to interact with humans in safety-sensitive situations (e.g autonomous vehicles, automated weapons, ...). In this context, having guarantees of reliability of the autonomous agents is of paramount importance.

The proposed controller was tested in inviscid simulations and on the VPM-2D simulator. It was shown to outperform the classical PD approach in the presence of disturbances and noise and to fit well to the VPM-2D simulator without any modification of the design. Finally the limitations of the controller were identified, such as instabilities arising with passivity violation (actuator saturation), or the sensitivity to initial condition (due to aggressive input profile). In practice the observed discrepancies may be mitigated by tuning properly the state penalty matrix, at the cost of less energy-efficiency.

This Chapter successfully solves the inner-loop problem to control the joints and achieve trajectory tracking in joint space. The question now arising is how to achieve trajectory tracking in the x, y -plane, i.e steering the body variables to a desired configuration. This question, related to outer-loop control, is addressed in the last Chapter.

Chapter 6

Guidance

A simple guidance algorithm is presented in this chapter to close the loop on the controller developed previously and achieve straight-line path following. The algorithm is tested in the inviscid simulator, with a SPR controller running in inner loop.

6.1 Line-Of-Sight Guidance

A simple gait generator based on the developments in [38] is implemented. The gait is responsible for giving the joints a desired angle ϕ_i^* to be followed by the inner-loop controller presented in Chapter 5. A typical sinusoidal gait is given as follows:

$$\phi_i^*(t) = \alpha g(i, N) \sin(\omega t + (i - 1)\delta) + \phi_0, \quad (6.1)$$

where $g(i, N)$ is the gait pattern, α the amplitude of oscillation, ω the frequency, δ a constant that adjusts phase shifts between the different segments and ϕ_0 the turning gait parameter (more on this later). For anguilliform swimming, typically we have: $g(i, N) = \frac{N-i}{N+1}$.

The motivation for these bio-inspired desired oscillations ϕ_i^* arises from the following biological observation: eels are propagating oscillation waves of increasing amplitude down the spinal cord with a constant phase delay δ between segments. The subsequent muscle contractions are responsible for propelling the eel forward [76]. When turning, eels produce the same rhythmic pattern, but biological observations pinpointed an important increase ϕ_0 in the curvature of their body on the turning side [76].

On top of that gait generator, a heading controller based on Line-Of-Sight (LOS) guidance is used to issue the control variable ϕ_0 proportional to the error between the desired and actual heading. This variable is used as an offset in the desired joint angle generation in order to achieve turning gait. The heading control law is:

$$\phi_0 = k_\theta(\psi - \bar{\theta}), \quad (6.2)$$

where k_θ is a fine tuned gain, $\bar{\theta}$ is the heading of the robot (in the present case, the head segment heading was taken as a proxy for the global heading) and ψ a desired heading obtained using LOS guidance to achieve straight-line following as illustrated in Figure 6.1 and given by the formula below:

$$\psi = -\tan^{-1}\left(\frac{p_y}{\Delta}\right), \quad (6.3)$$

where p_y is the cross-track error between the robot and the straight-line path, and Δ is a look-ahead parameter taken as a rule of thumb twice larger than the length of the robot.

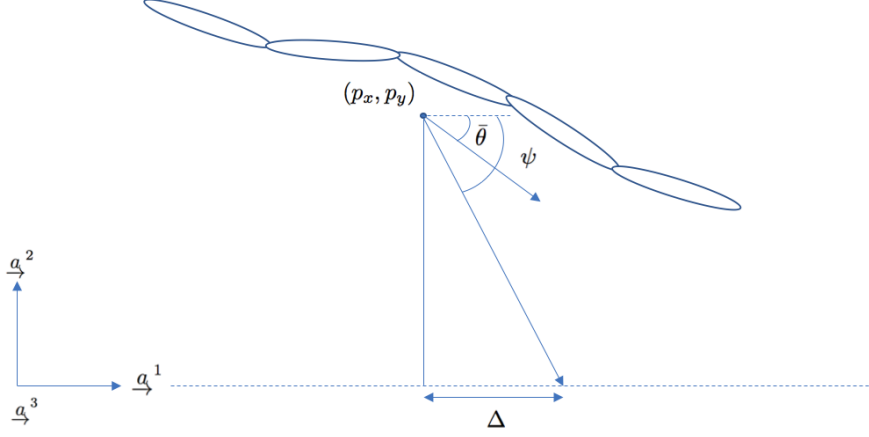


Figure 6.1: Principle of LOS guidance: the heading of the system is steered toward the look-ahead target.

The heading controller and gait generator described above are known as the outer-loop control block or guidance block. It feeds the inner control block with desired joint space trajectories. The entire control loop is shown in Figure 6.2.

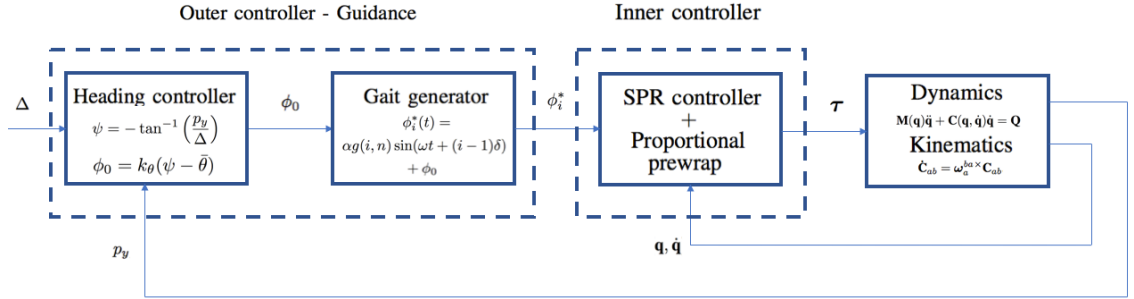


Figure 6.2: Closed loop: outer loop responsible for path following (issuing a desired trajectory in joint space); inner loop for achieving the desired trajectory.

6.2 Simulation Results

The values of the parameters chosen for the simulation are collected in Table 6.1. The influence of these parameters is abundantly discussed in the literature, e.g in [38]. They were obtained similarly in the present case by trial and errors, based on their physical interpretation. The geometric characteristics of the swimmer are collected in Table 6.2, using the usual notations (see Figure 3.1)

The simulation results for straight-line following are shown in Figure 6.3, where the x, y -position of the first segment in Figure 3.1 is plotted. On this Figure, the system

Table 6.1: Guidance parameters used in the simulation

N	5
k_θ	1
Δ (m)	4
ω (deg/s)	50
α (deg)	75
δ (deg)	30
$\mathbf{q}(0)$	$[0 \ 3 \ 0 \ 0 \ 0]^T$

Table 6.2: Geometric parameters of the eel-like swimmer

a (m)	0.2
b (m)	0.02
c (m)	0.04

is steered toward a straight-line path (aligned with $\vec{a}_1$ in Figure 6.1), starting with an initial cross-track error of $p_y = 3$ m and with an initial heading direction parallel to the path. As for any LOS algorithm (they are widely used in a variety of platforms, such as unmanned aerial vehicles or underwater unmanned vehicles), the performances are tributary of the look-ahead distance parameter: a large look-ahead distance yields an overdamped slow convergence, while a small look-ahead distance generates an oscillatory and fast convergence. Here the right trade-off is found with not too much oscillations. However, an initial undershoot is inevitable in that case, causing the robot to initially go in the wrong direction as seen. An initial undershoot and an overshoot of approximately 0.25 m (8% overshoot) are observed.

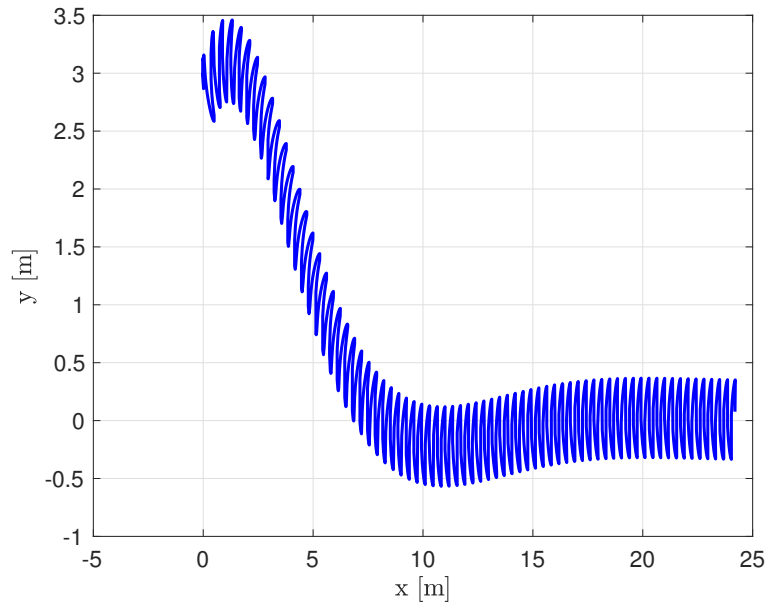


Figure 6.3: Straight-line following, starting with a cross-track error of $p_y = 3$ m and with an initial heading direction aligned with the path.

These results compare well with results obtained in the literature, see for example [38].

These result should however be mitigated by the slow convergence time (not shown on the graph). The best we could achieve with this algorithm is a settling time of 100 s. A similar result is reported in the literature. Note that the speed of the swimmer may be increased by increasing properly the amplitude and frequency of the sinusoidal gait generator in (6.1).

Also, the present algorithm only allows to achieve a straight-line following (extensions should be sought to achieve curved path tracking).

This is not really the kind of performances expected from an eel-like swimmer: swifter and faster motions would be expected.

6.3 Conclusion

The problem of guidance is addressed in this Chapter and solved with a classical LOS guidance algorithm combined with a bio-inspired sinusoidal gait pattern generator. The closed-loop is shown to achieve straight-line following with initial cross track error of $p_y = 3$ m. The performances obtained are similar to what has been achieved in the literature.

Although the obtained results are good in the sense that they demonstrate the scalability and validity of the controller developed in Chapter 5, the proposed approach is not quite satisfying for two reasons: it does not allow curved path following or trajectory tracking, and it does not yield fast enough convergence.

In order to improve these two issues in future work, some new cutting-edge researches are proposed: for example the use of Deep Reinforcement Learning (DRL) for finding the energy efficient patterns in swimming as reported in [8] or dynamical optimization proposed by the Extremum-Seeking (ES) guidance paradigm, see [77]. In the first approach, a deep neural net was trained in the event of cooperative swimming between two fish: one of the fish swims in the wake of a leading fish, aiming at exploiting vortices created in the flow for propulsion. After the fish has been trained on many simulations, it is able to reproduce the guidance scheme associated with the optimal energy-efficient behavior observed among natural fish. The second paradigm is concerned with ES control, which seeks a solution for an optimization problem in real time. ES control can be adapted to the guidance step in order to solve a dynamic optimization problem involving an unknown performance index: e.g in the context of anguilliform swimming, the aim of the guidance controller may be to minimize in real time an energy function so that minimum-energy trajectories are found, exploiting information from the environment such as pressure gradient or flow speed.

Approaches based on CPG as discussed in the first Chapter are also of particular interest for building robust oscillatory patterns [20].

Finally, additional simulation results are expected in order to test this algorithm on the VPM-2D simulator. The simulation times for guidance are however much higher than for inner-loop control.

Conclusion

This thesis was concerned with robust passivity-based control of an eel-like swimmer.

The motivation for a robust controller arose from these two observations:

- The difficulty of obtaining a workable and reliable model of the plant. Indeed, modeling the exact hydrodynamic fluid-structure interaction is a complex process requiring sophisticated numerical methods (for solving Navier-Stokes equations). Unfortunately, though reliable, these methods are untractable when it comes to a real time implementation on a micro-controller for the purpose of controlling the real system. By contrast, simplified models are workable solutions for both control synthesis (using them as a simulation environment) and control execution (use the simplified model of the plant in real time), but they are by essence "non-reliable" as they fail to capture the full complexity of the process.
- The wide variety of external disturbances acting on the plant. Ranging from eddies and vortices present in the flow, to streams and waves diverting the robot away from its goal, the role of external disturbances in underwater applications is significant and may cause important alterations of the performances.

In that respect, designing a controller that is robust to both unmodelled dynamics and external inputs is a highly desired achievement, and the direction taken in this work was to focus on the inner-loop control for achieving robust trajectory tracking in joint space with disturbance rejections. A consistent part of this work was also concerned with the derivation of a serious model of an eel-like swimmer, as the initial objective was to do "model-based" control. The final result – a SPR/VSP robust controller – thus reflects this dual objective as it demonstrates robust stabilization in the presence of disturbances, while being a model-based approach fitting optimally with the process (through the passivity framework and also from the optimal LQR-inspired synthesis).

Beyond this schematic view on the work achieved so far, many other results were obtained in this thesis and are summarized below:

- Experimental tests were carried out with the platform developed by former master's student. The test concluded that the robot was not suited to be used as an autonomous agent as it was suffering from stability issues and lack of controllability. The tests allowed nevertheless to give some recommendations on further work for improving the robot: replace the motors by torque controllable actuators, use reliable torque sensors, improve the stability, etc. Moreover, this test-bench – though not perfect – was sufficient for acquiring the data required in the validation of the simulator.

- The VPM-2D simulator, a high fidelity CFD code developed at Université Catholique de Louvain, was validated against experimental results. The trajectory in the x, y -plane predicted by the fluid solver was shown to correspond qualitatively with the trajectory achieved by the real robot.
- A localization system based on camera vision and color detection was implemented and validated. The system was however only used offline in post-processing.
- A model of an eel-like swimmer, based on the perfect fluid and irrotationality assumptions is established and validated against simulation results and literature results. The model treats the fluid-structure interaction as an addition to the inertia of the body. The derivation is inspired from an existing result in the literature, but the original result is extended in the Euler-Lagrange framework, instead of relying on the math-heavy Lagrange-Poincare approach. Comparisons with results in the literature showed that the model is consistent with respect to the original approach from which it was derived. Moreover, comparisons with VPM-2D allowed to conclude that the model presented is a good proxy for an eel-like swimmer, in a qualitative sense. As such, it was decided to use the proposed inviscid simulator as a tool for developing a controller.
- The passivity properties of the plant were unveiled from the inviscid model established, and a passivity based-approach was thus used: as passivity theorem states that a passive plant may be stabilized robustly by a Very Strictly Passive (VSP) controller, our interest focused on a linear Strictly Positive Real (SPR) controller – whose VSP-ness is guaranteed by the fundamental Kalman-Yakubovich-Popov (KYP) lemma. The synthesis of the SPR controller is done in an optimal way using a linearization of a model of the plant (actually a sub-part of the plant), then performing a LQR-inspired design. Simultaneously, the SPR-ness/VSP-ness assumptions are verified via KYP lemma. As an original contribution, the present work thus extends the SPR control approach to underactuated systems, and applies for the first time this kind of robust technique to underwater robotics. Simulation results showed the good robustness properties of the closed-loop, identifying however some drawbacks: on the one hand, altered performances are observed in some problematic situations where passivity violations may occur (e.g in case of actuator saturation) and on the other hand, a relative sensitivity to initial disturbances is identified. The word "relative" is employed because it was shown that by changing the penalty on the state (using an LQR analogy), the latter problem can be alleviated.

The advantages of the SPR approach are numerous: robustness and stability, optimality (in the sense of the LQR paradigm), scalability (automatic synthesis for any number of segments), offline synthesis (yielding low computational effort), intuitive tunability, etc.

- a guidance algorithm was implemented to close the loop on the controller designed and the closed-loop system was shown to achieve straight-path following. However, the performances obtained – though being comparable to what is achieved in the literature – are far below what nature can achieve.

From the results obtained so far, future work should focus on the following points:

- Implement the guidance on the VPM-2D simulator.

- Improving the guidance. The results obtained are not satisfying enough from the perspective of what is achieved in nature. CPG-based methods should be investigated. Methods relying on recent advances in Deep Reinforcement Learning are also promising, see the work in [8] where the optimal locomotion behavior is learned from training a deep neural net in efficient-swimming situations.
- Implement a gain-scheduled SPR/VSP controller. Indeed, the SPR/VSP-synthesis is based on a linearization of the plant about an operating points. Possibilities exist to implement gain-scheduled controllers linearized at many operation points and interpolated in real time by a scheduling signal. The passivity theorem would still guarantee the robust stability of the closed loop as the network interconnection of many VSP systems can be shown to still be VSP.
- Compliant and energy harvesting-based control approaches should also be investigated: for example, control techniques exploiting interesting nonlinearities in the flow (such as vortices in the wake of obstacles or other swimmers) could be developed in order to achieve energy efficient swimming. Energy harvesting through damping elements coupled with the actuators is discussed in [56], and vortice-based swimming in [78].

Appendices

Appendix A

Programming VPM-2D

The solver VPM-2D is implemented in C++ and the code is available on demand to the research team.

The file `StudyCaseDaRobotFish.h` located in the `source/studyCases` folder can be modified to specify physical parameters such as the density of the fluid and body, the fluid viscosity, the dimensions of the links, the number of links, and other simulation parameters such as the terminal time or the mesh refinement.

In the context of this thesis, the main part of the algorithm to focus on is concerned with the MBS solver. Indeed, one of the objective of this thesis is to design a control law issuing the relevant actuation torques. In order to modify the MBS step to include a custom controller, one has to write the `SegmentedFish.h` file located in the `source/shapes` folder of the VPM-2D package. In particular, the "methods" (functions) `rungeKutta`, `updateInSpace` and `fPrime` are useful: the two first functions allows to perform integrations if required (e.g for the computation of the integral term of a Proportional-Integral-Derivative controller, PID; or for more advanced dynamical controllers) while the latter allows to directly implement a control law by modifying the actuation torques. It is also worthwhile mentioning the "constructor" `SegmentedFish` that allows to specify initial conditions for the generalized coordinates.

When implementing any controller on the simulator, it should be expressed in a discretized form first. For controllers given in a Linear Time Invariant (LTI) state-space realization (as is the case in Chapter 5), a Zero Order Hold (ZOH) discretization method can be used, for example. Let the following state-space realization of a continuous-time LTI system:

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t), \quad (\text{A.1})$$

$$\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t). \quad (\text{A.2})$$

The ZOH discretization converts the above system into its discrete-time analogon via the following transformations [79]:

$$\mathbf{\Phi} = \exp(\mathbf{A}T), \quad (\text{A.3})$$

$$\mathbf{\Gamma} = \int_0^T \exp(\mathbf{A}\alpha)\mathbf{B}d\alpha, \quad (\text{A.4})$$

where T is the discretization step and the discrete-time formulation is given by:

$$\mathbf{x}[k+1] = \mathbf{\Phi}\mathbf{x}[k] + \mathbf{\Gamma}\mathbf{u}[k], \quad (\text{A.5})$$

$$\mathbf{y}[k+1] = \mathbf{C}\mathbf{x}[k] + \mathbf{D}\mathbf{u}[k]. \quad (\text{A.6})$$

Appendix B

Derivation of the Jacobian, Jacobian Transpose and Jacobian Derivative

The body- i -related Jacobian $\mathbf{J}_i$ is a matrix relating the absolute-frame-resolved twist of body i with the generalized coordinates:

$$\xi_{i_a} = \dot{\mathbf{r}}_{i_a} = \sum_{k=1}^n \frac{\partial \mathbf{r}_{i_a}}{\partial q_k} \dot{q}_k = \mathbf{J}_i(\mathbf{q}) \dot{\mathbf{q}}, \quad (\text{B.1})$$

Reasoning on the equation above, it appears that each element $(\xi_{i_a})_j$, $\forall j$ taken independently is the sum of the n generalized coordinates weighted by the n terms $\frac{\partial (r_{i_a})_j}{\partial q_k}$, $k = \{1, \dots, n\}$, $\forall j$. These terms are thus linked to the Jacobian by:

$$(J_i)_{jk} = \frac{\partial (r_{i_a})_j}{\partial q_k} \quad (\text{B.2})$$

which provides a practical way of computing the Jacobian componentwise by direct differentiation of $\mathbf{r}_{i_a}$.

By a quick trigonometric consideration, the pose of the centroids of all considered links resolved in $\mathcal{F}_a$ is given as follows:

$$\mathbf{r}_{i_a} = \begin{bmatrix} x + L \sum_{l=1}^{i-1} \cos(\mathcal{Q}(l)) + \frac{L}{2} \cos(\mathcal{Q}(i)) \\ y + L \sum_{l=1}^{i-1} \sin(\mathcal{Q}(l)) + \frac{L}{2} \sin(\mathcal{Q}(i)) \\ 0 \\ 0 \\ 0 \\ \mathcal{Q}(i) \end{bmatrix}. \quad (\text{B.3})$$

We recall that $\mathcal{Q}(\alpha) = \theta + \sum_{\delta=1}^{\alpha-1} \phi_\delta$.

Deriving the pose in a fashion dictated by equation B.2, the following non-zero components of the Jacobian are obtained:

$$J_{11} = \frac{\partial (r_{i_a})_1}{\partial x} = 1, \quad (\text{B.4})$$

$$J_{22} = \frac{\partial(r_{i_a})_2}{\partial y} = 1, \quad (\text{B.5})$$

$$J_{1k} = \frac{\partial(r_{i_a})_1}{\partial q_k} = -L \sum_{l=1}^{i-1} \sin(\mathcal{Q}(l)) \mathcal{Q}(l)^{(q_{k-2})} - \frac{L}{2} \sin(\mathcal{Q}(i)) \mathcal{Q}(i)^{(q_{k-2})}, \quad k > 2, \quad (\text{B.6})$$

$$J_{2k} = \frac{\partial(r_{i_a})_2}{\partial q_k} = L \sum_{l=1}^{i-1} \cos(\mathcal{Q}(l)) \mathcal{Q}(l)^{(q_{k-2})} + \frac{L}{2} \cos(\mathcal{Q}(i)) \mathcal{Q}(i)^{(q_{k-2})}, \quad k > 2, \quad (\text{B.7})$$

$$J_{6k} = \frac{\partial(r_{i_a})_6}{\partial q_k} = \mathcal{Q}(l)^{(q_{k-2})}, \quad k > 2, \quad (\text{B.8})$$

all other components zero. The term $\mathcal{Q}(\alpha)^{(q_{k-2})} = \frac{\partial \mathcal{Q}(\alpha)}{\partial q_{k-2}}$ is equal to 1 if $2 < k < \alpha + 2$, and 0 otherwise.

Equations (B.4) – (B.8) can be readily used to algorithmically generate the Jacobian. The transpose of the Jacobian follows straightforwardly from the above equations.

The derivative of the Jacobian with respect to the m -th generalized coordinate is also of valuable use. It can be obtained by derivation of equations (B.4) – (B.8). The non zero terms are ($k > 2$, $m > 2$):

$$\left(\frac{\partial J_{1k}}{\partial q_m} \right)_{1k} = -L \sum_{l=1}^{i-1} \cos(\mathcal{Q}(l)) \mathcal{Q}(l)^{(q_{k-2}, q_{m-2})} - \frac{L}{2} \cos(\mathcal{Q}(i)) \mathcal{Q}(i)^{(q_{k-2}, q_{m-2})}, \quad (\text{B.9})$$

$$\left(\frac{\partial J_{2k}}{\partial q_m} \right)_{2k} = -L \sum_{l=1}^{i-1} \sin(\mathcal{Q}(l)) \mathcal{Q}(l)^{(q_{k-2}, q_{m-2})} - \frac{L}{2} \sin(\mathcal{Q}(i)) \mathcal{Q}(i)^{(q_{k-2}, q_{m-2})}, \quad (\text{B.10})$$

all other terms zero. The term $\mathcal{Q}(\alpha)^{(q_{k-2}, q_{m-2})} = \frac{\partial^2 \mathcal{Q}(\alpha)}{\partial q_{m-2} \partial q_{k-2}}$ is equal to 1 if $2 < k < \alpha + 2$ and $2 < m < \alpha + 2$, 0 otherwise.

Equations (B.9) and (B.10) can be readily used to algorithmically generate the Jacobian derivative.

Appendix C

LaSalle's Invariant Set Theorem

LaSalle's invariant set theorem is a classical result used when the traditional Lyapunov's second method does not allow to conclude on asymptotic stability (i.e when the time derivative of the Lyapunov candidate function is negative semi-definite). LaSalle's invariant set theorem provides a guarantee of convergence of the system trajectories toward a so-called invariant set under some assumptions. One of the important assumptions is that the system be autonomous (i.e not submitted to any external inputs), which is typically the case for closed-loop systems with initial conditions. First, the following definitions of stability are recalled:

Definition 8 (Lyapunov Stability and Asymptotic Stability). *Consider the following autonomous system:*

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t)), \quad \mathbf{x}_0 = \mathbf{x}(0), \quad (\text{C.1})$$

with an equilibrium at $\mathbf{x}_e$ such that $\mathbf{f}(\mathbf{x}_e) = 0$. Then:

- *The equilibrium point is said to be Lyapunov stable if, $\forall \epsilon > 0$, $\exists \delta_1 > 0$ such that, if $\|\mathbf{x}_0 - \mathbf{x}_e\| < \delta_1$, then $\forall t \geq 0$ one have $\|\mathbf{x}(t) - \mathbf{x}_e\| < \epsilon$.*
- *The equilibrium point is said to be asymptotically stable if it is Lyapunov stable and $\exists \delta_2 > 0$ such that if $\|\mathbf{x}_0 - \mathbf{x}_e\| < \delta_2$, then $\lim_{t \rightarrow \infty} \|\mathbf{x}(t) - \mathbf{x}_e\| = 0$.*

The illustration of the Lyapunov stability concept, using the notations above, is shown in Figure C.1.

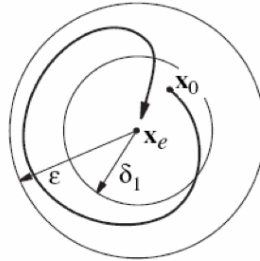


Figure C.1: Lyapunov stability.

LaSalle's theorem for autonomous systems is now stated [71]:

Theorem 5 (LaSalle's Invariant Set Theorem). *Consider the following autonomous system:*

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t)), \quad \mathbf{x}_0 = \mathbf{x}(0), \quad (\text{C.2})$$

and let Ω be a compact positively invariant set with respect to (C.2), i.e $\mathbf{x}(0) \in \Omega \Rightarrow \mathbf{x}(t) \in \Omega \quad \forall t \in \mathbb{R}^+$.

Let $\mathcal{V}$ be a continuously differentiable function such that, along the trajectories of (C.2):

$$\dot{\mathcal{V}}(\mathbf{x}(t)) \leq 0, \quad \forall \mathbf{x}(t) \in \Omega. \quad (\text{C.3})$$

Let:

$$E = \{\mathbf{x}(t) \in \Omega \mid \dot{\mathcal{V}}(\mathbf{x}(t)) = 0\}, \quad \forall \mathbf{x}(t) \in \Omega. \quad (\text{C.4})$$

Let M be the largest invariant set with respect to (C.2) contained in E . Then, every solution of (C.2) starting in Ω approaches M as $t \rightarrow \infty$.

Bibliography

- [1] M. Gazzola, P. Chatelain, W. M. Van Rees, and P. Koumoutsakos, “Simulations of single and multiple swimmers with non-divergence free deforming geometries,” *Journal of Computational Physics*, vol. 230, no. 19, pp. 7093–7114, 2011.
- [2] G. Vanderwegen and M. Barbason, *Swimming eel-like robot*. Ecole polytechnique de Louvain, Université catholique de Louvain, Prom.: Ronsse, Renaud and Chatelain, Philippe, 2017.
- [3] D. Lachat, A. Crespi, and A. J. Ijspeert, “Boxybot: a swimming and crawling fish robot controlled by a central pattern generator,” in *Biomedical Robotics and Biomechatronics, 2006. BioRob 2006. The First IEEE/RAS-EMBS International Conference on*. IEEE, 2006, pp. 643–648.
- [4] B. Bayat, A. Crespi, and A. Ijspeert, “Envirobot: A bio-inspired environmental monitoring platform,” in *Autonomous Underwater Vehicles (AUV), 2016 IEEE/OES*. Ieee, 2016, pp. 381–386.
- [5] P. Liljebäck and R. Mills, “Eelume: A flexible and subsea resident imr vehicle,” in *OCEANS 2017-Aberdeen*. IEEE, 2017, pp. 1–4.
- [6] R. Pfeifer, M. Lungarella, and F. Iida, “Self-organization, embodiment, and biologically inspired robotics,” *science*, vol. 318, no. 5853, pp. 1088–1093, 2007.
- [7] H. Hu, “Biologically inspired design of autonomous robotic fish at essex,” in *IEEE SMC UK-RI Chapter Conference, on Advances in Cybernetic Systems*. Citeseer, 2006, pp. 3–8.
- [8] S. Verma, G. Novati, and P. Koumoutsakos, “Efficient collective swimming by harnessing vortices through deep reinforcement learning,” *Proceedings of the National Academy of Sciences*, p. 201800923, 2018.
- [9] J. M. Anderson and N. K. Chhabra, “Maneuvering and stability performance of a robotic tuna,” *Integrative and comparative biology*, vol. 42, no. 1, pp. 118–126, 2002.
- [10] Z. Wang, Q. Gao, and H. Zhao, “Cpg-inspired locomotion control for a snake robot basing on nonlinear oscillators,” *Journal of Intelligent & Robotic Systems*, vol. 85, no. 2, pp. 209–227, 2017.
- [11] A. J. Ijspeert, A. Crespi, and J.-M. Cabelguen, “Simulation and robotics studies of salamander locomotion,” *Neuroinformatics*, vol. 3, no. 3, pp. 171–195, 2005.
- [12] A. J. Ijspeert, “Central pattern generators for locomotion control in animals and robots: a review,” *Neural networks*, vol. 21, no. 4, pp. 642–653, 2008.

- [13] F. Boyer, D. Chablat, P. Lemoine, and P. Wenger, "The eel-like robot," in *ASME 2009 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. American Society of Mechanical Engineers, 2009, pp. 655–662.
- [14] C. M. Breder, "The locomotion of fishes," *Zoologica*, vol. 4, no. 5, pp. 159–297, 1926.
- [15] V. van Ginneken, E. Antonissen, U. K. Müller, R. Booms, E. Eding, J. Verreth, and G. van den Thillart, "Eel migration to the sargasso: remarkably high swimming efficiency and low energy costs," *Journal of Experimental Biology*, vol. 208, no. 7, pp. 1329–1335, 2005.
- [16] J. F. Vincent, O. A. Bogatyreva, N. R. Bogatyrev, A. Bowyer, and A.-K. Pahl, "Biomimetics: its practice and theory," *Journal of the Royal Society Interface*, vol. 3, no. 9, pp. 471–482, 2006.
- [17] O. Holland, "Grey walter: the pioneer of real artificial life," in *Proceedings of the 5th international workshop on artificial life*. MIT Press, Cambridge, 1997, pp. 34–44.
- [18] A. J. Ijspeert, A. Crespi, D. Ryczko, and J.-M. Cabelguen, "From swimming to walking with a salamander robot driven by a spinal cord model," *science*, vol. 315, no. 5817, pp. 1416–1420, 2007.
- [19] J. Ayers, J. Witting, S. Ryder, and C. Olcott, "Process and architecture of robotic system to mimic animal behavior in the natural environment," Aug. 3 2010, uS Patent 7,769,487.
- [20] A. J. Ijspeert, "Central pattern generators for locomotion control in animals and robots: a review," *Neural networks*, vol. 21, no. 4, pp. 642–653, 2008.
- [21] J. L. Krichmar, D. A. Nitz, J. A. Gally, and G. M. Edelman, "Characterizing functional hippocampal pathways in a brain-based device as it solves a spatial memory task," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 102, no. 6, pp. 2111–2116, 2005.
- [22] R. Pfeifer and G. Gómez, "Morphological computation—connecting brain, body, and environment," pp. 66–83, 2009.
- [23] S. Nolfi and M. Mirolli, *Evolution of communication and language in embodied agents*. Springer Science & Business Media, 2009.
- [24] R. Pfeifer, F. Iida, and G. Gomez, "Designing intelligent robots," *Journal of the Robotics Society of Japan*, vol. 24, no. 7, pp. 783–790, 2006.
- [25] M. Lungarella and O. Sporns, "Mapping information flow in sensorimotor networks," *PLoS computational biology*, vol. 2, no. 10, p. e144, 2006.
- [26] A. Vardy and R. Moller, "Biologically plausible visual homing methods based on optical flow techniques," *Connection Science*, vol. 17, no. 1-2, pp. 47–89, 2005.
- [27] A. D. Horchler, R. E. Reeve, B. Webb, and R. D. Quinn, "Robot phonotaxis in the wild: a biologically inspired approach to outdoor sound localization," *Advanced Robotics*, vol. 18, no. 8, pp. 801–816, 2004.

- [28] F. Hara and R. Pfeifer, *Morpho-functional machines: The new species: Designing embodied intelligence*. Springer Science & Business Media, 2003.
- [29] B. Bayat, N. Crasta, H. Li, and A. Ijspeert, “Optimal search strategies for pollutant source localization,” in *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*. Ieee, 2016, pp. 1801–1807.
- [30] K. Abhishek, “An autonomous robotic fish for mobile sensing.”
- [31] J. Liu, I. Dukes, and H. Hu, “Novel mechatronics design for a robotic fish,” in *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on*. IEEE, 2005, pp. 807–812.
- [32] M. S. Triantafyllou and G. S. Triantafyllou, “An efficient swimming machine,” *Scientific american*, vol. 272, no. 3, pp. 64–70, 1995.
- [33] R. Mason and J. Burdick, “Construction and modelling of a carangiform robotic fish,” pp. 235–242, 2000.
- [34] R. Mason and J. W. Burdick, “Experiments in carangiform robotic fish locomotion,” in *Robotics and Automation, 2000. Proceedings. ICRA’00. IEEE International Conference on*, vol. 1. IEEE, 2000, pp. 428–435.
- [35] A. J. Ijspeert and J. Kodjabachian, “Evolution and development of a central pattern generator for the swimming of a lamprey,” *Artificial life*, vol. 5, no. 3, pp. 247–269, 1999.
- [36] A. D. Marchese, C. D. Onal, and D. Rus, “Autonomous soft robotic fish capable of escape maneuvers using fluidic elastomer actuators,” *Soft Robotics*, vol. 1, no. 1, pp. 75–87, 2014.
- [37] W. Stoll, “Airacuda, a bionic object with a fin drive,” *Festo A. G.*, 2010.
- [38] E. Kelasidi, P. Liljebäck, K. Y. Pettersen, and J. T. Gravdahl, “Innovation in underwater robots: biologically inspired swimming snake robots,” *IEEE Robotics & Automation Magazine*, vol. 23, no. 1, pp. 44–62, 2016.
- [39] A. Crespi and A. J. Ijspeert, “Amphibot ii: An amphibious snake robot that crawls and swims using a central pattern generator,” in *Proceedings of the 9th international conference on climbing and walking robots (CLAWAR 2006)*, no. BIOROB-CONF-2006-001, 2006, pp. 19–27.
- [40] M. Porez, F. Boyer, and A. J. Ijspeert, “Improved lighthill fish swimming model for bio-inspired robots: Modeling, computational aspects and experimental comparisons,” *The International Journal of Robotics Research*, vol. 33, no. 10, pp. 1322–1341, 2014.
- [41] T. Takayama and S. Hirose, “Amphibious 3d active cord mechanism" helix" with helical swimming motion,” in *Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on*, vol. 1. IEEE, 2002, pp. 775–780.
- [42] S. Hirose and H. Yamada, “Snake-like robots [tutorial],” *IEEE Robotics & Automation Magazine*, vol. 16, no. 1, pp. 88–98, 2009.
- [43] R. Fong and A. J. Ijspeert, “Envision, a vision system for envirobot,” 2018.

- [44] T. Paschal, “Undulatory swimming vs propeller based.”
- [45] E. Kanso, J. E. Marsden, C. W. Rowley, and J. B. Melli-Huber, “Locomotion of articulated bodies in a perfect fluid,” *Journal of Nonlinear Science*, vol. 15, no. 4, pp. 255–289, 2005.
- [46] G. S. Chirikjian, “Hyper-redundant manipulator dynamics: A continuum approximation,” *Advanced Robotics*, vol. 9, no. 3, pp. 217–243, 1994.
- [47] E. Kelasidi, K. Y. Pettersen, and J. T. Gravdahl, “A control-oriented model of underwater snake robots,” in *Robotics and Biomimetics (ROBIO), 2014 IEEE International Conference on*. IEEE, 2014, pp. 753–760.
- [48] P. Liljebäck, K. Y. Pettersen, Ø. Stavdahl, and J. T. Gravdahl, “A review on modelling, implementation, and control of snake robots,” *Robotics and Autonomous Systems*, vol. 60, no. 1, pp. 29–40, 2012.
- [49] E. Kelasidi, P. Liljebäck, K. Y. Pettersen, and J. T. Gravdahl, “Integral line-of-sight guidance for path following control of underwater snake robots: theory and experiments,” *IEEE Transactions on Robotics*, vol. 33, no. 3, pp. 610–628, 2017.
- [50] A. M. Kohl, E. Kelasidi, K. Y. Pettersen, and J. T. Gravdahl, “Model-based los path-following control of planar underwater snake robots,” pp. 343–363, 2017.
- [51] F. Delcomyn, “Neural basis of rhythmic behavior in animals,” *Science*, vol. 210, no. 4469, pp. 492–498, 1980.
- [52] A. Crespi, D. Lachat, A. Pasquier, and A. J. Ijspeert, “Controlling swimming and crawling in a fish robot using a central pattern generator,” *Autonomous Robots*, vol. 25, no. 1-2, pp. 3–13, 2008.
- [53] A. Billard and A. J. Ijspeert, “Biologically inspired neural controllers for motor control in a quadruped robot,” in *Neural Networks, 2000. IJCNN 2000, Proceedings of the IEEE-INNS-ENNS International Joint Conference on*, vol. 6. IEEE, 2000, pp. 637–641.
- [54] Z. Wu, J. Yu, M. Tan, and J. Zhang, “Kinematic comparison of forward and backward swimming and maneuvering in a self-propelled sub-carangiform robotic fish,” *Journal of Bionic Engineering*, vol. 11, no. 2, pp. 199–212, 2014.
- [55] L. Righetti and A. J. Ijspeert, “Programmable central pattern generators: an application to biped locomotion control,” in *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*. IEEE, 2006, pp. 1585–1590.
- [56] C. Bernier, R. Ronsse, M. Gazzola, and P. Chatelain, “Simulations of propelling and energy harvesting articulated bodies via remeshed vortex methods,” *ArXiv e-prints*, vol. 1805.10691, 2018.
- [57] Robotis. Dynamixel: Ax-12a. [Online]. Available: <http://www.crustcrawler.com/products/bioloid/docs/AX-12.pdf>
- [58] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, “A density-based algorithm for discovering clusters in large spatial databases with noise,” *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96)*, pp. 226–231, 1996.

- [59] RoboSavvy. Ax-12 measures. [Online]. Available: <https://robosavvy.com/forum/viewtopic.php?t=3363#p20230>
- [60] R. Musgrave, M. Moghaddami, V. Avsarkisov, R. Wang, and W. Zhu, “Point vortex dynamics in two dimensions,” *Spring School on Fluid Mechanics and Geophysics of Environmental Hazards*, 2009.
- [61] P. C. Hughes, *Spacecraft Attitude Dynamics*, 2nd ed. Mineola, New York: Dover, 2004.
- [62] M. W. Spong, S. Hutchinson, M. Vidyasagar *et al.*, *Robot modeling and control*. Wiley New York, 2006, vol. 3.
- [63] H. Lamb, *Hydrodynamics*. Cambridge university press, 1932.
- [64] N. Newman, *Marine Hydrodynamics*. MIT Press, 1977.
- [65] J. D. Eldredge, “Dynamically coupled fluid–body interactions in vorticity-based numerical simulations,” *Journal of Computational Physics*, vol. 227, no. 21, pp. 9170–9194, 2008.
- [66] A. Van der Schaft and A. Van Der Schaft, *L2-gain and passivity techniques in nonlinear control*. Springer, 2000, vol. 2.
- [67] J. R. Forbes, “Extensions of input-output stability theory and the control of aerospace systems,” Ph.D. dissertation, 2011.
- [68] M. Vidyasagar, *Nonlinear systems analysis*. Siam, 2002, vol. 42.
- [69] A. Teel, T. Georgiou, L. Praly, and E. Sontag, “Input-output stability,” *The control handbook*, vol. 1, pp. 895–908, 1996.
- [70] J. Wyatt, L. Chua, J. Gannett, I. Goknar, and D. Green, “Energy concepts in the state-space theory of nonlinear n-ports: Part i-passivity,” *IEEE transactions on Circuits and Systems*, vol. 28, no. 1, pp. 48–61, 1981.
- [71] H. K. Khalil, “Nonlinear systems,” *Prentice-Hall, New Jersey*, vol. 2, no. 5, pp. 5–1, 1996.
- [72] J.-J. E. Slotine, W. Li *et al.*, *Applied nonlinear control*. Prentice hall Englewood Cliffs, NJ, 1991, vol. 199, no. 1.
- [73] A. Walsh and J. R. Forbes, “A very strictly passive gain-scheduled controller: Theory and experiments,” *IEEE/ASME Transactions on Mechatronics*, vol. 21, no. 6, pp. 2817–2826, 2016.
- [74] G. Slater and M. McLaren, “Robust multivariable control of large space structures using positivity,” *Journal of Guidance, Control, and Dynamics*, vol. 10, no. 4, pp. 393–400, 1987.
- [75] A. J. Rojas, “On the discrete-time algebraic riccati equation and its solution in closed-form,” *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 162–167, 2011.
- [76] K. A. McIsaac and J. P. Ostrowski, “Motion planning for anguilliform locomotion,” *IEEE Transactions on Robotics and Automation*, vol. 19, no. 4, pp. 637–652, 2003.

- [77] A. Walsh, J. R. Forbes, S. A. Chee, and J. J. Ryan, “Kalman-filter-based unconstrained and constrained extremum-seeking guidance on so (3),” *Journal of Guidance, Control, and Dynamics*, pp. 1–12, 2017.
- [78] D. Beal, F. Hover, M. Triantafyllou, J. Liao, and G. Lauder, “Passive propulsion in vortex wakes,” *Journal of Fluid Mechanics*, vol. 549, pp. 385–402, 2006.
- [79] K. J. Åström and B. Wittenmark, *Computer-controlled systems: theory and design*, 3rd ed. Dover, 2013.

