

EU FP7



AMARSi

Adaptive Modular Architectures for Rich Motor Skills

ICT-248311

D 3.3

September 2012 (30 months)

In silico case studies of compliant robots

Authors: Harold Roberto Martinez Salazar (UZH)- Chapter: 1, 3
Qian Zhao (UZH)- Chapter: 1, 2, 6
Juan Pablo Carbajal (UZH)- Chapter: 1, 3, 4
Hidenobu Sumioka (UZH)- Chapter: 2, 6
Xiaoxiang Yu (UZH)- Chapter: 2, 6
Kohei Nakajima (UZH)- Chapter: 2, 6
Rolf Pfeifer (UZH)- Chapter: 1, 2, 6
Andrea d'Avella (SLF)- Chapter: 4
Cristiano Alessandro(UZH)- Chapter: 4
Yuri Ivanenko (SLF)- Chapter: 3
Ken Caluwaerts (UGENT)- Chapter: 1, 5
Michiel Dhaene (UGENT)- Chapter: 5
David Verstraeten(UGENT)- Chapter: 5
Benjamin Schrauwen(UGENT)- Chapter: 5

Due date of deliverable	1st September 2012
Actual submission date	28th September 2012
Lead Partner	UZH
Revision	Final
Dissemination level	Public

Contents

1	Introduction	4
2	The Function of the Spine and its Morphological Effect in Quadruped Robot Locomotion	8
2.1	Introduction	8
2.2	Design	9
2.2.1	Spinal morphology design	9
2.2.2	Controller design	10
2.3	Simulation	12
2.3.1	Simulation Setup	12
2.3.2	Overall exploration based on two spinal morphologies	12
2.3.3	Basic effects of thoracic joint	13
2.3.4	Dynamic locomotion induced by double flight phase	15
2.4	Discussion	16
2.5	Conclusion and Outlook	17
3	Robustness: a new criterion for gait transitions in bipedal locomotion based on the SLIP model	21
3.1	Introduction	21
3.2	Methods	23
3.2.1	Simulation of the dynamics	23
3.2.2	Viability, Robustness and symmetric gaits	24
3.3	Results	25
3.3.1	Conditions for transitions	26
3.3.2	Mechanism of gait transitions	27
3.3.3	Qualitative Prediction of Biomechanical Observables	29
3.3.4	Robust Hopping Gait	32
3.4	Discussion	33
3.5	Conclusion	36

4	Synthesis and Adaptation of Effective Motor Synergies for the Solution of Reaching Tasks	38
4.1	Introduction	38
4.2	Definitions and Methods	39
4.2.1	Solution to point-to-point reaching tasks	40
4.2.2	Synthesis and Development of Synergies	42
4.3	Results	43
4.4	Discussion	44
4.5	Conclusion and Future Work	48
5	Locomotion without a brain: Physical Reservoir Computing in Tensegrity structures	49
5.1	Introduction	49
5.2	Tensegrity structures	52
5.2.1	Tensegrity dynamics	52
5.2.2	Related work on tensegrity structures	55
5.3	Central pattern generators	56
5.3.1	Matsuoka oscillators	57
5.4	Physical Reservoir Computing	58
5.5	Experiments	61
5.5.1	Outsourcing motor pattern generation	62
5.5.2	Applications	69
5.5.3	The importance of complex dynamics	73
5.6	Discussion	75
5.7	Conclusions	78
6	Embodiment Enables the Spinal Engine in Quadruped Robot Locomotion	80
6.1	Introduction	80
6.2	Experimental setting	81
6.2.1	Quadruped robot	81
6.2.2	Information theoretic measures	84
6.3	Experiments	86
6.3.1	Versatile behaviors	86
6.3.2	The effect of spinal morphology on the bounding and trotting gaits	89
6.3.3	Information theoretical analysis on the bounding gait	89
6.4	Conclusion and discussion	93
A	Supplement material of Chapter 2	96

Chapter 1

Introduction

In the deliverable 3.2 we presented how the morphological computing approach can significantly facilitate the control strategy in several scenarios, e.g. quadruped locomotion, bipedal locomotion and reaching. In particular, the Kitty experimental platform is an example of the use of morphological computation to allow quadruped locomotion. In this deliverable we continue with the simulation studies on the application of the different morphological computation strategies to control a robotic system.

Low-level

In chapter 2 we introduce the development of a SimMechanics® simulation of the Kitty platform used to perform a deep study of the effects of the spine-driven locomotion behaviors. For this study, we developed quadruped models featuring one and two spinal joints. We chose three individuals from these two models and analyzed their behaviors in terms of gait properties, i.e. angle of attack, ground clearance, and movement of the center of mass. The results show that employing the spinal morphology with two joints can greatly enhance the stability and speed of locomotion. Among several advantageous properties of the two spinal joint model we identify the following: First, it allows the robot to adjust the movement of the center of mass to stabilize itself. Second, by providing more freedom to bend the spine the robot can pull the rear legs forward, thus increasing the stride length. Finally, locomotion with this model exhibits two flight phases per stride and a low gait duty factor, similar to what it is observed in running cheetahs.

In the same vein, in chapter 3 we used the Spring-Loaded Inverted Pendulum (SLIP) model together with a novel robustness criterion to explain smooth transitions between running and walking at constant energy. The robustness of a gait can be understood as the attentional demand required

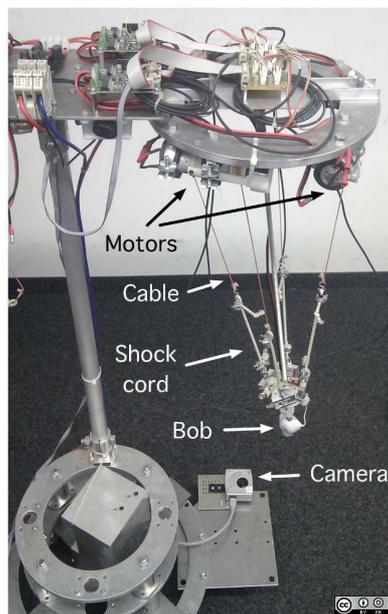


Figure 1.1: Tendulum robot: tendon driven pendulum robot

to maintain it. If highly precise inputs are needed to continue with a gait the system must spend more resources to select an adequate action (e.g. use of detailed models, better estimation of states from noisy sensory data, more processing time; i.e. cognitive load or attention). We show that gait transitions in the SLIP model, executed under the robustness criterion, satisfactorily predicts several biomechanical indicators such as Froude number, hip excursion, gait duty factor and vertical ground reaction force profiles. A consequence of this study is that the development of a transition can be explained as a result of the trade-off between robustness and energetic cost. Summarizing, our study shows that the SLIP model is able to reproduce several biomechanical indicators of human locomotion and strikingly, the controller can be imprecise.

Chapter 4 of this deliverable reports our efforts to bring together the results presented in chapter 2, 10 and 11 of deliverable 3.2. That is, we start to bring together numerical results in morphological computation into the control realm and into real experimental platforms, and at the same time, we shorten the distance to biomechanical hypothesis of motor control. Taking inspiration from the hypotheses of muscle synergies, we present a method to generate open loop controllers for an agent solving point-to-point reaching tasks. The controller output is defined as a linear combination of a small set of predefined actuations (synergies). The method can be interpreted from

a developmental perspective, since it allows the agent to autonomously synthesize and adapt an effective set of synergies to new behavioral needs. This scheme greatly reduces the dimensionality of the control problem, while keeping a good performance level. Currently, we are working to apply the method in the Tendulum robot (tendon driven pendulum robot, see Fig. 1.1). Applying it to this device implies that the method cannot rely on the existence of an exact mathematical model, as it is believed to be in biological agents.

Chapter 5 introduces full morphological control of compliant and redundant tensegrity structures. Tensegrity structures are similar to mass-spring systems, but can be free-standing by the inclusion of compressive elements. These structures can be used to model biomechanical systems at different scales. By analyzing this extreme instantiation of compliant structures, we demonstrate the existence of a spectrum of choices on how to implement control in the body-brain composite. We build upon the foundations of D 3.1 and demonstrate the practical feasibility of the abstract methods introduced by Hauser and Maass. Online learning rules are applied to show that it is possible to outsource the generation of complex (CPG-like) motor patterns to the body itself and that external feedback can intrinsically be integrated in the control loop. This is achieved by training static, linear feedback controllers, based on the Reservoir Computing principles, for the non-linear tensegrity structures. Since the various linear learning rules we consider differ in biological plausibility, and no specific assumptions are made on how to implement the feedback in a physical system, there is room to adapt the learning rules to a specific robot platform (e.g. the tendulum).

High-level

In chapter 6, we have extended the study of morphological computation on Kitty (spine-driven quadruped robot) such that the robot can extract high-level notions of itself from low level sensory data. The Kitty platform has been used to demonstrate versatile behaviors (bounding, trotting, and turning) through the coupling between the controller, the body and the environment in D3.2 (see deliverable 3.2)). In this study we use information theoretic analysis to characterize the structure of the sensorimotor data generated during the bounding gait. We analyze three different spinal morphologies by changing the position of the virtual spinal joint under the same control parameters. As a result, the velocity of locomotion of kitty is affected drastically by the change of the virtual joint. The highest velocity is produced when the virtual joint is on the rear position and the lowest velocity is produced when the virtual joint is on the front side. Furthermore, we show that the information transfer and the association between sensor values and

motor signals is strongly affected by the spinal morphology. When the virtual joint is on the rear side of the robot the information transfer is the highest, and when the virtual joint is on the front side of the robot the information transfer is the lowest. Based on these results, we show how low level sensory data can be used to identify the spinal morphology, and the velocity of the robot. We also started to apply this strategy to recognize different kinds of terrain. So far, Kitty without touch sensors on the feet can use the sensory data from the spine to recognize whether the robot is on the ground.

Chapter 2

The Function of the Spine and its Morphological Effect in Quadruped Robot Locomotion

2.1 Introduction

Legged robotics has drawn much more attention from robotic researchers due to its applications in rough terrains in nature and in our living environments [1]. Most of the existing quadruped robots are very similar in their morphology, and feature a single rigid body with four legs with individually actuated hips and/or knees. However, the resulting locomotion behavior is much more constrained than its natural counterpart in terms of speed, energy efficiency, maneuverability, and adaptivity to rough terrain.

From a biological point of view, one of the major differences between robots and animals is the spine. It is central to the control of body posture, provides the foundation to produce the leg's movement, and integrates limb and trunk actions [2]. Principally, quadruped animals use rhythmic movements of the body stem with its axial skeleton and legged locomotion strategies in parallel [3]. For example, a cheetah, the fastest animal in the land, is able to reach up to 110 km/h for a short dash. The main role of its spine is to make extensive body articulation, thus leading to greater power and speed.

Nevertheless, there have only been a few attempts to introduce a spine to a robotic platform, while substantial effort has been put on the design and optimization of leg's morphology and its associated controller [4–6]. Recently, some researchers have come to realize the important role the spine plays in locomotion, but most only focused on the controller of the spinal joint, and

barely paid attention to its morphology [7–9]. All the aforementioned studies simply introduced a spinal joint connecting the fore and rear part without further study on its morphological parameters and anatomical structure, e.g., the position/number of the joints.

The concept of embodiment suggests that a system’s behavior is generated through the interaction between controller, body (morphology) and environment [10]. A system even without a controller is able to generate versatile and meaningful behavior. For example, a new study has demonstrated how the arrangement of springs located in the spine generates and affects locomotion behavior of going down a slope without external energy in a passive quadruped robot [11]. If we look back to the anatomical structure of a biological spine, we find more important spinal morphological parameters need to be investigated further, except spinal stiffness, to gain a deep understanding of its underlying mechanism.

In this paper, we introduce two spinal morphologies differing in the number of spinal joints into a quadruped model to demonstrate the spine-driven locomotion behavior. Three individuals from these two spinal morphologies are selected and compared regarding the gait, the attack angle, the ground clearance (GC), and the movement of the center of mass (CoM). The simulation results show that the locomotion can be greatly enhanced by employing lumbosacral joint and thoracic joint together in terms of the stability and speed.

2.2 Design

In this section, we describe the design of the spinal morphologies and its associated models. Next, the selection of the morphological parameters and the design of controller are presented.

2.2.1 Spinal morphology design

The spine is made up of small bones, known as vertebrae, that are stacked on top of each other to create the spinal column. The number of vertebrae varies with the species of the animals from ten in frogs to fifty six in tigers. All of the spinal movements are distributed over the connecting joints of these vertebrae. So it is too complex to analyze the spine-driven locomotion by taking many joints into account.

We know that the spinal column consists of lumbosacral spine, thoracic spine, and cervical spine [12]. So we employed a spinal joint with one degree of freedom to emulate the movement of each part. Since the head’s movement

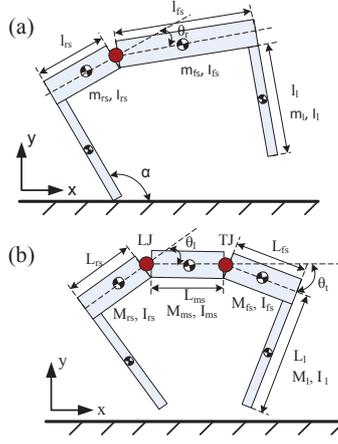


Figure 2.1: Planar quadruped model of M2. Red dots stand for the actuated spinal joints. The specifications of the model are shown in Table 2.1.

has less effect on locomotion, we ignored cervical spine in this paper. We employ a lumbosacral joint (LJ) to mimic the role of lumbosacral spine in locomotion. Similarly, the thoracic joint (TJ) is taken to emulate the function of thoracic spine.

As a starting point, we only applied LJ into the model to study the role of lumbosacral spine, because lumbosacral spine’s main motions are bending and extending [13], which could greatly benefit locomotion. LJ is located in the rear part of the spine, inspired by the biological finding which suggests that the rear position of LJ can produce a particularly pronounced sagittal displacement of the pelvis [3]. We define the model with LJ as morphology one (M1). M1 consists of three segments which are a pair of stick-shaped legs, and a spine with a LJ. We simplified this model by taking out the leg actuation, to focus on the study of spine-driven locomotion and the effects of spinal morphology on locomotion.

Because of the existence of a small amount of flexion-extending movement in thoracic spine [14], we added a TJ in the middle between the shoulder and the LJ (Fig. 2.1) to investigate how it contributes locomotion, along with lumbosacral spine. The model with these two joints is named by morphology two (M2). If we fix the movement of TJ, which is θ_t in Fig. 2.1, and kept the rest parameters of M2, then M2 becomes M1.

We copied some of cheetah’s morphological parameters (weights and sizes of the body and legs), and applied them to the models, because a cheetah exhibits noticeable spinal flexion and extension movement when running[15]. Table 2.1 details morphological parameters we have chosen for M2.

2.2.2 Controller design

Minimalistic control strategy

We employed a minimalistic control strategy to this model [16], in which the angular position of the spinal joints is determined by the sinusoidal curve as

follows:

$$\theta_l(t) = A_l \sin(2\pi f_l t) + \psi_l \quad (2.1)$$

$$\theta_t(t) = A_t \sin(2\pi f_t t + \phi_t) + \psi_t \quad (2.2)$$

where θ_l and θ_t indicate the target angular positions of the motors controlling LJ and TJ, respectively. A , f and ψ designate the amplitude, the frequency, and the offset. The phase ϕ is the delay between the LJ and TJ. The subscript l and t denote LJ and TJ, respectively. By using this simple control scheme, we are able to evaluate how the morphological properties of the spine can contribute to locomotion behavior. The parameters used in the following experiments are heuristically determined as follows: $f_t = f_l = 1.5$, $A_l \in [17, 33]$, and $\psi_l \in [-12, 2]$. The rest control parameters (A_t , ψ_t , ϕ_t) will be optimized with genetic algorithm described in the following part. Time step t in this paper represents one actuation loop of the control program.

Genetic algorithm for the sinusoid function controller

The genetic algorithm (GA) is used to optimize the control parameters (A_t , ψ_t , ϕ_t) for TJ with the aim to achieve fast and stable locomotion behaviors. The population size is 60 while the number of generations is 10. The cost function is the speed multiplied by -1 . The probabilities of crossover and mutation are 0.5 and 0.15, respectively. Each individual consists of three parameters which are encoded as three 8 bit genes. The boundaries of these three parameters are decided heuristically as follows: $A_t \in [15, 25]$, $\psi_t \in [-4, 4]$, and $\phi_t \in [-1.5708, -0.7854]$.

Table 2.1: Morphological parameters for M2

Param.	Value	Param.	Value	Param.	Value
L_l	0.83 m	M_l	5 kg	I_l	0.2 kg · m ²
L_{rs}	0.33 m	M_{rs}	6.7 kg	I_{rs}	0.25 kg · m ²
L_{ms}	0.33 m	M_{ms}	6.7 kg	I_{ms}	0.25 kg · m ²
L_{fs}	0.33 m	M_{fs}	6.7 kg	I_{fs}	1.73 kg · m ²

L: length; M: weight; I: inertia.

l : leg; rs : rear spinal segment; ms : middle spinal segment; fs : fore spinal segment.

2.3 Simulation

In this section, the results of the overall exploration based on these two proposed models are presented first, followed by the selection of the best individual from M1, namely I1, and its comparison with the one from M2, namely I2. The latter has the same control parameters for the LJ as I1, to ensure fairness. Finally the best individual from M2, namely I3, is selected and analyzed in details.

2.3.1 Simulation Setup

We have implemented both models in Mathworks matlab 2009 (64bit), together with the SimMechanics toolbox.

In simulation, we constructed a physically realistic interaction model based on a biomechanical study [17]. The vertical ground reaction forces are modeled by one non-linear visco-elastic element, and the horizontal forces are calculated by a sliding-stiction model. It switches from stiction to sliding when the velocity of the foot exceeds the specified threshold. We used 0.7, 0.8, and 0.01 m/s for the sliding, stiction friction coefficients and the threshold velocity, respectively. Simulations were started from an initial condition with a height of 0.1 m from a stationary state and run for 50 s.

2.3.2 Overall exploration based on two spinal morphologies

To achieve comprehensive behavioral analysis, we investigated the influence of amplitude (A_l) and offset (ψ_l) on the locomotion behavior. We varied A_l from 17° to 33° , and ψ_l from -12° to 4° with the increment of 2° in M1. Then we keep the same control parameters for the LJ and optimize the rest three (A_t , ψ_t , ϕ_t) for the TJ in M2. These parameters, A_l and ψ_l significantly change the locomotion behavior: the robot exhibits a stable rapid locomotion; it runs slowly; it exhibits unstable behavior; or it falls over. In this paper, we use two methods together, the step-to-fall method and the apex return map, to judge the system's stabilizing behavior [18]. If the robot does not fall within 50 s and the error of two adjacent apex heights of the CoM is less than 0.15 m after initial transient, this run is considered to be successful and therefore the speed is recorded, otherwise it is a failure and the speed is set to 0 m/s.

Fig. 2.2 (a), (b) suggest that the locomotion is able to be generated by the spinal flexion and extension. M2 can move much faster than M1, and its best performance attains 2.3 m/s, while the best one from M1 is 0.63 m/s.

The attack angle is defined as the angle formed between the leg and the ground in the forward direction when the feet touch on the ground. In both morphologies, greater attack angle of rear legs (Fig. 2.2 (c), (d)) corresponds to faster speed (Fig. 2.2 (a), (b)). With a larger attack angle, the rear legs can rotate the robot’s body around the contact point and push it more forward.

In the biological perspective, the CoM moves forward and backward alternatively with respect to its nose during locomotion [19]. We define the CoM_S as the distance between the position of CoM and the position of the robot’s shoulder, instead of the nose. The range of the CoM_S gets wider, as a result of the increasing amplitude of the bending and extension movement (Fig. 2.2 (e), (f)). Wider range of the CoM_S (Fig. 2.2 (e), (f)) is associated with better performance (Fig. 2.2 (a), (b)), because it offers more freedom to adjust the CoM, benefiting the stabilization of the posture and the enhancement of the speed. Furthermore, with the increase of the speed in M1 and M2 (Fig. 2.2 (a), (b)), the values of the rear and fore boundaries of the CoM_S get smaller (Fig. 2.2 (g)-(j)), which means that the horizontal excursion of the CoM moves further to the anterior trunk region.

The function of GC is to overcome the obstacles. Higher GC (Fig. 2.2 (k)-(l)) corresponds to fast speed (Fig. 2.2 (a)-(b)). However, higher GC makes the robot unstable. It is easier to fall when the spinal movement is pronounced in M1, compared to M2. M2 is able to use an additional spinal joint (TJ) to reduce GC of fore legs, adjust the CoM, and stabilize the robot.

2.3.3 Basic effects of thoracic joint

To understand the underlying mechanism of the resulting different behaviors from M1 and M2, we analyzed the behavior of I1 from M1, which attains 0.63 m/s (Fig. 2.3 (d)), and I2 from M2, which attains 1.25 m/s, defined previously (Fig. 2.3 (e)). Parameters obtained from the genetic algorithm described in the previous section are given in Table 2.2.

Table 2.2: Optimal parameters for M2

M2	A_l	ψ_l	A_t	ψ_t	ϕ_t
I2	19°	-12°	16.4°	-3.2°	-0.9
I3	31°	-4°	23.8°	3°	-0.9

Analysis on spine-driven locomotion

Fig. 2.4 (a), (b) show that the stable locomotion behavior of I1 and I2 can be achieved, even if leg actuation is not taken into account. We observe that four phases exist in I1 (Fig. 2.3(g), (m)). Since the phase shaded by green has very short duration, low GC of fore legs with 0.016 m, and almost has the same posture as the one after it shaded by blue, we assign both to phase II. I1 is featured with three prominent phases as shown in Fig. 2.4(a). Starting from the original posture (phase I), the spine is flexed and the rear legs are pulled forward until the maximum (phase II). This moves the CoM forward. Afterwards, the spine is extended to allow the lift up of the fore legs, leading to the back-moving of the CoM (phase III). In the next step, the fore legs touch the ground, and the CoM moves forward again (back to phase I). The same process repeats.

Similarly as I1, I2 also has three important phases (2.4 (b)). The difference with I1 comes from the further flexed spine caused by combining the flexion of LJ and TJ. This then pulls the rear legs more forward than I1 (phase I) and leads to a higher attack angle of 123° (Fig. 2.3(l)), compared to I1 with 116° (Fig. 2.3(j)). The rest of the cycle follows the same procedure as in I1.

The period of one cycle of I2 (105 time steps per cycle) is longer than I1 (94 time steps per cycle), but the speed is much faster, due to the increase of the stride length caused by the combination of these two spinal joints' movements.

Ground clearance

GC for the fore legs is almost the same in I1 and I2. It has two peaks: one lower about 0.015 m, and the other one higher about 0.22 m. However, GC for the rear legs is different for I1 and I2. The former has its GC barely noticeable (0.003 m), while the latter has a much higher GC (0.01 m). This is due to the inclusion of the flexion of the additional spinal joint (TJ).

Attack angle

In these two spine-driven models, attack angle is changed along with the body posture controlled by the spinal controller.

Wider range of attack angle of fore legs in I2 enhances locomotion, because it is able to increase the stride length by propelling the body forward further. It varies from 94° to 74° in phase I (Fig.2.3 (k)), as a result of the additional flexion of TJ. Therefore, it can push the body forward further than I1, whose angle is almost constant, 90° (Fig.2.3 (j)). In addition, larger attack angle of

rear legs contributes to the increase of the stride length by pushing the body more forward.

Movement of the center of mass

Fig. 2.3 (a), (b) show that the horizontal motion of the CoM in the body is only determined by flexion and extension of the back. This underlines the determinant role of the spine as the main engine for the locomotion.

The movement of CoM relative to the shoulder is not constant (Fig. 2.3 (a), (b)). The horizontal excursion of the CoM is in coupling with the motion of the spine. During spinal extension, the CoM moves to the posterior part of the spine, but it moves to the anterior part during spinal flexion. This horizontal excursion equals about 4%, 4% of the model's length in I1, I2, respectively. The extension phase of the spine is coupled with a upward movement of the CoM. In the flexion phase, after initial ascent, the CoM moves downward (Fig. 2.3 (a), (b)). The excursion of the vertical movement of the CoM is about 14%, 16% of the model's length in I1, I2, respectively.

2.3.4 Dynamic locomotion induced by double flight phase

To know how well M2 is able to perform, we pick the fastest one from M2, namely I3, and compare it with I1 and I2. I3 can reach up to 2.3 m/s (Fig. 2.3 (f)) .

Analysis on spine-driven locomotion

There is a high degree of co-ordination between spinal flexion and the placing of the feet on the ground to maximize stride and increase speed in I3. I3 mainly differs from I1 and I2 in the gait. It is characterized by five phases, two of which are flight phases, instead of one, in each stride, as shown in Fig. 2.3 (i), (o). Fig. 2.4 (c) shows that one takes place when the spine is at maximum extension (phase I); the other one happens when sharp contraction of the spine takes place before the rear feet contact the ground (phase IV). The period of one cycle of I3 is the same as I2 and longer than I1, but speed is much faster than both. The double flight phases can account for its fast speed.

Ground clearance

I3 has pronounced GC not only for fore legs with 0.37 m, but for rear legs (Fig. 2.3 (i)) . It exhibits two flight phases in rear legs in each cycle: one is

with GC of 0.073 m and the other one is of 0.09 m, which are much larger than I1 and I2.

Attack angle

For the rear legs, it has similar value of attack angle to I2, but it has a smaller lift up angle of 80° , which can crouch more and push the body forward further, compared to I2 with the angle of 90° .

Movement of the center of mass

Fig. 2.3 (a), (b), (c) show that I3 has similar horizontal and vertical movement of the CoM to I1 and I2 during one cycle. This horizontal excursion equals around 6% and vertical excursion is about 20% of the model's length in I3.

Table. 2.3 shows the boundaries and the range of the CoM_S in horizontal and vertical direction. We observed that values of fore boundary and rear boundary of the horizontal movement of CoM_S in I2 and I3 are smaller than I1, which suggests that I2 and I3 are able to move the CoM forward more efficiently than I1, benefiting the rapid locomotion. Moreover, the excursion range of the CoM_S in I2 and I3 is wider than I1, offering more freedom to adjust the CoM to stabilize the robot itself.

2.4 Discussion

We noticed that the stride frequency of I2 and I3 is lower, which should have led to a slower speed. However, this is compensated by the increase of stride length, which is caused by the introduction of the additional TJ. As an overall effect, average speed of I2 and I3 is higher.

I2 is capable of producing more pronounced spinal movements, which contribute to the increase of the stride length by pulling the rear legs forward further than I1, thus increasing the stride length. I2 has less attack angle of

Table 2.3: Results of CoM_S in I1, I2, I3

	CoM- $S_{horizontal}$ (m)			CoM- $S_{vertical}$ (m)		
	Fore _b	Rear _b	Range	Low _b	High _b	Range
I1	0.46	0.5	0.04	0.25	0.12	0.13
I2	0.45	0.49	0.04	0.25	0.12	0.14
I3	0.42	0.48	0.06	0.29	0.11	0.18

fore legs when they lift off the ground, caused by additional TJ, which can propel the body forward. We believe that multiple spinal joints are able to provide the body with more freedom to enlarge the swing of the limbs and increase the stride length.

I2 and I3 mainly differ in the speed and the gait, as a result of the amplitude of spinal movements (Table. 2.2). I3 almost runs twice as fast as I2. It reaches the maximal extension and flexion in two flight periods per stride, while I2 is only suspended once in each stride. In addition, I3's gait exhibits greater proportion of flight in total stride. These results are consistent with studies of the motions of the running cheetah and horse [15]. A horse, with relatively rigid spine generating less spinal movements, can be represented by I2, and a cheetah, featuring with pronounced spinal movements, is suitable to be simplified as I3. We conclude that the double flight periods, together with greater proportion of flight, contribute to its longer stride [15]. However, I3 exhibits a double stance phase (phase III in Fig. 2.4 (c)), which does not exist in cheetah running. We could eliminate this phase by adding actuated hip joints. When the rear feet touch on the ground in phase II, the rear hip motor is actuated and the leg is swung outward. As a consequence, the body is propelled forward and the rear feet are off the ground in the next phase, which might avoid the presence of this double stance phase.

The horizontal excursion of CoM relative to shoulder equals about 4%, 4%, 6% of the model's length in I1, I2, I3, respectively. They are less than 15% from pika [19], which could be improved by introducing more spinal joints. The spinal joint in this sense can be defined as the connecting point of vertebrae in animals. A cat has thirty vertebrae in its spinal column, five more vertebrae than a human. This might account for its spine's agility and rapid speed.

The amplitude of the vertical motion relative to the nose is about 14%, 16%, 20% of the model's length in I1, I2, I3, respectively. This is higher than the the average value of 10 % observed from human [20] and pika [19] running. The reduction of the vertical displacement of the CoM could be achieved by introducing springs in the legs and adjusting their spring-mass systems by increasing the angle swept by the stance legs while keeping leg stiffness nearly constant [21] [22].

2.5 Conclusion and Outlook

This novel study suggested that the motion of the spine is a determinant factor in the locomotion. The change of spine posture serves the placement

of the CoM relative to the ground contact point, working as an engine to propel the body; limbs might be looked at as servants of the trunk to assist locomotion [12].

M2 performs better than M1 in terms of the speed and stability. M2 is able to produce more freedom to pull the rear legs forward, increase the stride length, and move the CoM more efficiently forward. Therefore the speed is increased. In addition, it benefits stability by using additional TJ to optimize the movement generated by the LJ by readjusting unstable posture or enhancing the extension-flexion movement. I3, the best individual from M2, outperforms I2 due to its double flight phases and greater proportion of flight in total stride, as a result of more pronounced spinal movement. This is similar to what we observe from the cheetah's running, which makes significant difference in the speed and gait.

In the future, compliant and actuated legs will be introduced to study how to reduce vertical excursion of the CoM. In addition, the way of how to coordinate legs' movement and the spine's movement would be another direction.

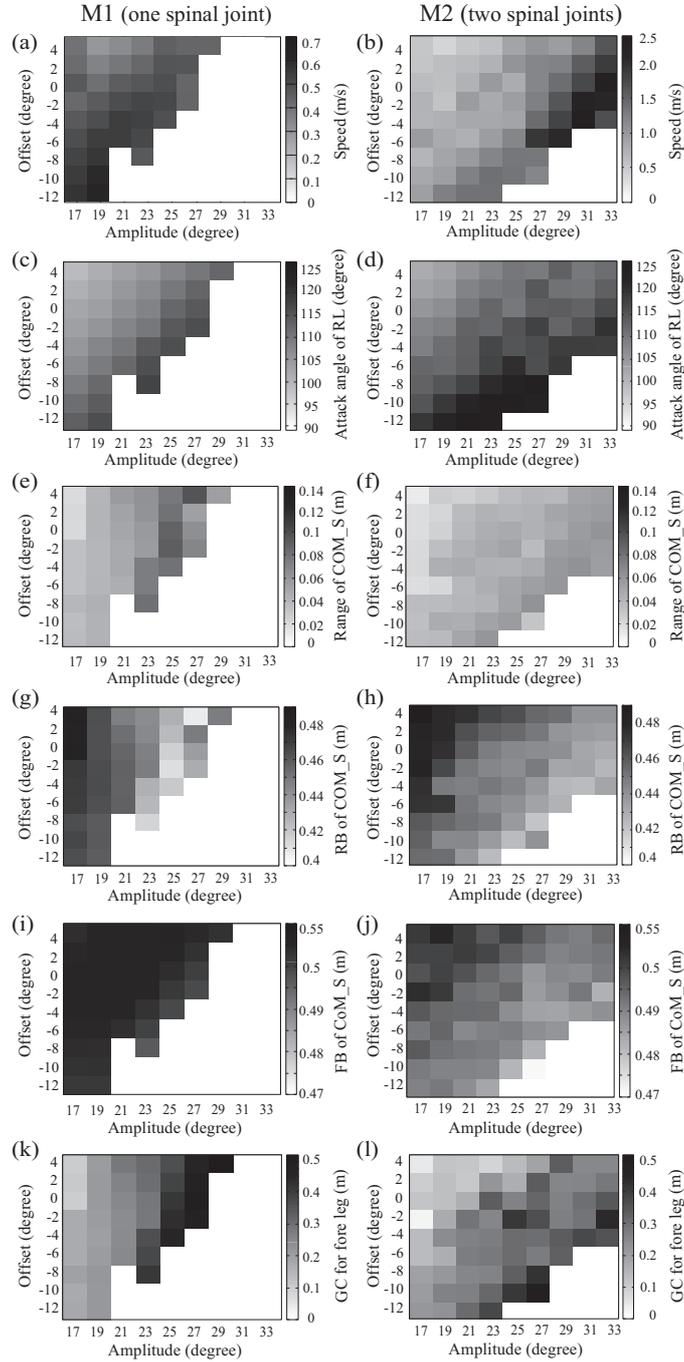


Figure 2.2: Comparison results of two spinal morphologies (M1 in the left column and M2 in the right column). X axis is amplitude (A_l), and y axis denotes offset (ψ_l) for the LJ. The intensity of the cell represents the speed in (a), (b); the attack angle of real legs (RL) in (c), (d); the range of CoM_S in (e), (f); the rear boundary (RB) of CoM_S in (g), (h); the fore boundary (FB) of CoM_S in (i), (j); the ground clearance of fore legs (k), (l).

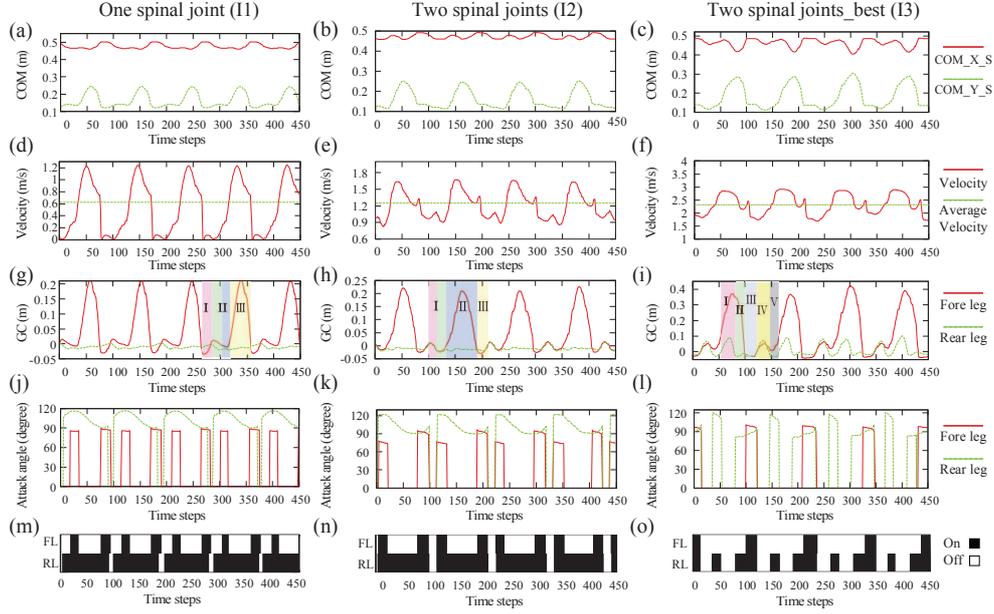


Figure 2.3: The left, middle, right column are the results of I1, I2, and I3, respectively. The x axis represents time steps. The y axis stands for the movement of CoM relative to the shoulder (a), (b), (c); the velocity (c), (d), (e); the height of ground clearance (g), (h), (i); and the attack angle (j), (k), (l). In (g), (h), (i), areas shaded stand for phases, consistent with phases marked in Fig. 2.4. The footfall patterns of I1, I2, I3, are represented in (m), (n), and (o).

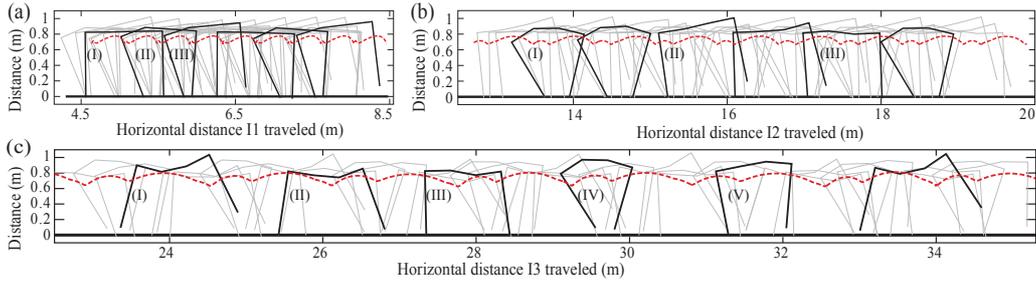


Figure 2.4: Stick figures illustrating three different behaviors in simulation. The body postures are illustrated every 94 and 658 (94×7) simulation steps (a), and every 105 and 630 (105×6) simulation steps (b), (c) (gray and black stick figures, respectively). Red dotted line represent the trajectories of absolute CoM. (a) I1 ($A_l = 19^\circ$, $\psi_l = -12^\circ$). (b) I2 ($A_l = 19^\circ$, $\psi_l = -12^\circ$). (c) I3 ($A_l = 31^\circ$, $\psi_l = -4^\circ$).

Chapter 3

Robustness: a new criterion for gait transitions in bipedal locomotion based on the SLIP model

3.1 Introduction

The study of bipedal locomotion has motivated the development of several models that explain the most important principles governing the dynamics of the gaits. Some researchers have adopted models that include detailed representations of the different leg components (e.g. neuromuscular structures) or physical elements that emulate the neuromuscular structures such as springs, dampers and multi-segmented legs ([23–26]). Although these models reproduce the dynamics of locomotion, they are not widespread use as conceptual models because of their complexity.

In contrast, simpler models have been used extensively as conceptual models of bipedal locomotion[27, 28]. Most of these models were developed to explain the exchange of kinetic and potential energy of the center of mass (CoM) of biological agents. During walking, kinetic and potential energy of the CoM are out of phase, i.e. the maximum height of the CoM corresponds with a minimum of its speed[29, 30]. In consequence, the inverted pendulum (IP) model [27, 31] is frequently used to represent walking, since in this model the exchanges of energy are also out of phase. This model uses a rigid rod with mass to represent the leg and a point mass on top of it to represent the body. Detailed analyses of the passive dynamics of this model have shown interesting mathematical features such as stable one-period gaits,

two-periods gaits and bifurcations leading to chaos [32]. These discoveries constituted a conceptual cornerstone for the development of passive dynamic walkers, i.e. mechanical machines which are capable of stable locomotion without any actuators or controllers [33]. Despite its conceptual explanatory power, the IP model does not correctly reproduce several aspects of human walking [34], e.g. the vertical oscillations of the CoM experimentally observed are smaller than the ones predicted by the model. Moreover, this model does not reproduce the dynamics of gait transitions and over estimates the speed in which a transition from walking to running is induced[35].

Running is commonly represented with another model, the spring-loaded inverted pendulum (SLIP) [28, 36]). The SLIP model consist of a point mass (the body) attached to a massless spring (the leg). During the stance phase the spring is fixed to the ground via an ideal revolute joint that is removed during flight phase. This model has been successfully used for the control of running machines [37–39]. In contrast with the IP model, the SLIP model can be extended to reproduce the mechanics of human walking (this is achieved by adding another massless spring modeling the second leg). However, the analyses carried out with this model had not yet explained gait transitions at the same energy, e.g. from walking to running at a characteristic Froude number. Previous findings suggested that transitions were possible only by drastically increasing or decreasing the total energy to induce a considerable change in the forward speed of the system [40].

Given that the legs in the SLIP model are massless, their swinging motion cannot be directly described using equations derived from Newton’s laws. Therefore, a control policy that sets the angle of attack at touchdown (the angle spanned by the landing leg and the horizontal at the time the foot collides with the ground) must be defined a priori. Generally, the angle of attack at touchdown is kept constant. Herein, we assume a more general control policy: the system selects a new angle of attack at each step. However, this selection is considered imprecise, e.g. the selected angle might be subject to small fluctuations. Under this mildly adverse conditions we still identify regions of the phase portrait of the system where it is capable of taking an infinite number of steps (we call them robust regions) and regions where the system can change from one gait to another (called transition regions).

In this study, we propose robustness as a criterion to explain the onset of gait transitions, complementing the classical energetic criterion [41, 42]. The robustness of a gait can be understood as the attentional demand required to maintain it. If highly precise inputs are needed to continue with a gait the system must spend more resources to select an adequate action, e.g. use of detailed models, better estimation of states from noise sensory data, more processing time; i.e. cognitive load or attention. This new perspective is ac-

accompanied with a trade-off between robustness and energetic cost. A similar trade-off have been observed in bees[43]. When flying in turbulent flows, the animal extends its lower limbs reducing the chances of rolling, but increases the drag force sacrificing forward speed. Furthermore, the transitions found under the newly included robustness criterion qualitatively reproduce experimental values of the changes in the amplitude of the oscillations of the hip, changes in the gait duty factor and variations of ground reaction forces. Incidentally, these transitions use a gait pattern that we identify with hopping (a variation of the homonyms gait described in [44]).

This paper is organized as follows. In section 3.2, we describe the models used for the simulation and introduce the necessary concepts. In section 3.3 we show the regions of robust locomotion and gait transition. In that section we also compare our results with biological data. Discussions are given in section 3.4 and we conclude the paper in section 3.5.

3.2 Methods

The time evolution of a gait is segmented in several phases, each phase is described with a sub-model. These sub-models represent the motion of a point mass under the influence of: only gravity (flight phase), gravity and a linear spring (single stance phase), gravity and two linear springs (double stance phase). The point mass stands for the body of the agent and the massless linear springs model the forces from the legs. During stance, the springs are fixed to the ground via a revolute joint, i.e. we assume that the friction is enough to prevent foot slipping. A trajectory switches from one sub-model to another when certain *event functions* [45–47] cross zero. These functions indicate events such as touchdown, liftoff and they also indicate the failure of the system, i.e. the agent falls to the ground or it moves backwards. During walking, running and hopping the system always goes through the single stance phase, therefore all gaits can be studied and compared during this phase. We study the system using a return map (see [48], ch. 8.7), i.e. we observe the system when the support leg reaches a vertical orientation, that is, when the angle between the horizontal and the leg is equal to 90° .

3.2.1 Simulation of the dynamics

As we said above, the state of the system is observed when a trajectory intersects a section, called \mathcal{S} , defined by the support leg forming a right angle with the ground. At this section the state of the system is defined by the height of the hip (i.e. height of the CoM), r , and the velocity in the vertical

direction, v_y (v_x is obtained from the equation of constant energy [45]). This intermittent observation of the system renders the continuous evolution of the model into a mapping that transforms states in the section at time t to states in the section at $t + \Delta t$. Under this representation running is a map $\mathcal{R}_\alpha : \mathcal{S} \rightarrow \mathcal{S}$ that transforms points through the evolution of the system from the single stance phase to the flight phase and back again to the single stance phase using an angle of attack α . Similarly, the walking map $\mathcal{W}_\alpha : \mathcal{S} \rightarrow \mathcal{S}$ transforms points through the evolution of the system from the single stance phase to the double stance phase and back again to the single stance phase using only one angle of attack α .

All initial conditions are given in the \mathcal{S} section and in the single stance phase, i.e. only one leg touching the ground and oriented vertically. All possible pairs (r, v_y) are simulated for a given value of the total energy E in the range [780, 900]J at intervals of 10 J. The model implementation is carried out in MATLAB(2009, The MathWorks). Simulations are run using the step variable integrator ode45 (relative tolerance: 1×10^{-6} and absolute tolerance: 1×10^{-8}). Experimental data analysis was done using GNU Octave [49].

3.2.2 Viability, Robustness and symmetric gaits

Viability, as presented in [45], defines the easiness of taking a further step. That is, the wider the range of angles of attack that can be used to take a step the easier is to take that step. In a physical platform it is required that the angle of attack exists for a definite interval, since real sensors and actuators have a finite resolution and are affected by noise. A viability region of the section \mathcal{S} are all the states that can be mapped once (i.e. taking a step) selecting an angle of attack from an interval of reasonable length. We use the symbol $V^i(\Delta\alpha)$ to denote the viability region of gait i with minimum angle interval $\Delta\alpha$. The superindex i denotes the gait and can be any of $\{\mathcal{R}, \mathcal{W}\}$. For example, the region where a running system can choose a viable angle of attack in an interval of two degrees or more is $V^{\mathcal{R}}(2^\circ)$.

The concept of *robustness* is defined on top of that of viability. A robust state of the section \mathcal{S} is a viable state that can always be mapped into a viability region by choosing the right sequence of angles of attack. Each angle should be selected from an interval of at least length $\Delta\alpha$. This assumes that the controller can select an angle of attack for each step. In particular, some of the self-stable regions identified in [40] belong to a robust region. However, this does not mean that the system remains in the self-stable region for each step (that would imply that the angles of attack is selected with enormous precision), but that when the system lies in that region at time t , it can

remain close to it by selecting angles with finite resolution. The robust regions of the section \mathcal{S} are denoted with the symbol $P^i(\Delta\alpha)$, where the superindex i indicates the corresponding gait.

The gaits commonly used by humans are symmetric, meaning that the dynamical behavior of the left leg mirrors the one of the right leg. In our model, this is possible when two conditions are satisfied: the velocity in the vertical direction at \mathcal{S} is zero and there is an angle of attack α that can bring the system back to the same state.

3.3 Results

We report the results obtained from the study of gait transitions in the SLIP model following the criterion of robustness detailed in Section 3.2.2. It turns out that the concept of robust gaits offer an alternative explanation for the onset of gait transitions in bipedal locomotion, comparable with arguments based on metabolic costs.

We show that the discovery of robustness as a useful criterion to induce gait transitions allows for qualitative comparisons with experimental biomechanical data. In particular we present results in terms of *Froude number*, *hip excursion*, *gait duty factor*, and *vertical ground reaction forces*. The Froude number is the ratio between the weight and the centripetal force $w^2 l_o / g$, where g is the acceleration due to gravity, l_o is the natural length of the leg and w is the angular velocity of the body around the foot in contact with the ground. Hip excursion denotes the amplitude of vertical oscillations of the hip. The gait duty factor is the fraction of the total duration of a gait cycle in which a given foot is on the ground. The vertical ground reaction force is vertical component of the normal force exerted by the ground.

We begin our exposition with a detailed explanation of the conditions, in terms of decrease of robustness, that may trigger gait transitions. From there we move on to describe the mechanism underlying robust gait transitions. The results of those two sections are combined to present qualitative comparison with biomechanical observables, followed by a short description of robust hopping.

Figure 3.1 shows the range of forward speed for robust gaits at several energies. Evidently, robust walking exists only at low locomotion energies, while running increases robustness for higher energies. These observations are consistent with the experimental results reported in [50], where it was shown that imposed fast walking required higher attention than running at similar speeds. Furthermore, normal switching between gaits did not required high attentional demand. Details are given in subsequent sections.

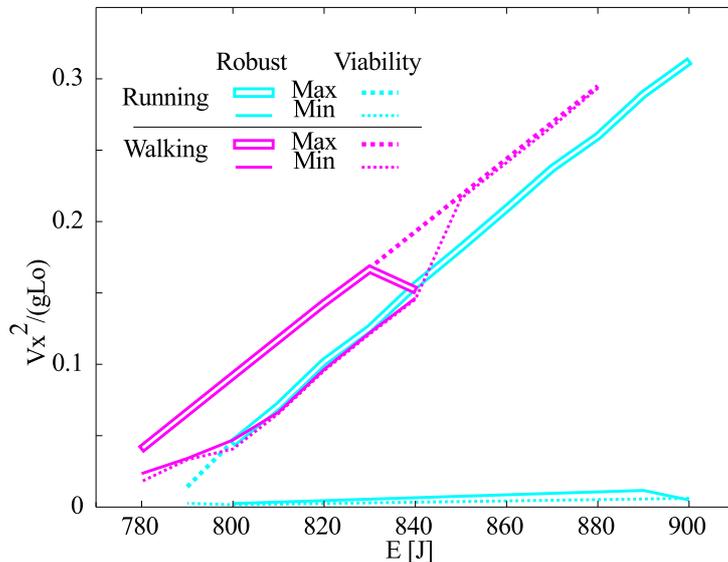


Figure 3.1: (Color online) Horizontal velocity as a function of the total mechanical energy of the system. The forward velocity in the section \mathcal{S} is used to estimate the Froude number of symmetric gaits for different energies. The dotted lines show the maximum (big dots) and minimum (small dots) Froude number corresponding to the state in the viability region. The lines show the maximum (outline) and minimum (solid) Froude number corresponding to the state in the robust region. The (blue) light gray color represents \mathcal{R} and the (magenta) dark gray color represents \mathcal{W} .

3.3.1 Conditions for transitions

The definition of robust gait applies for symmetric and non-symmetric gaits. Figure 3.2 shows the robust regions, $P^i(2^\circ)$, and the viability regions, $V^i(2^\circ)$, for energies between 790 J, and 850 J. Since all the possible states of the system lie in a hemispherical region (see equations (15)-(21) of [45]), we marked the apex of this hemisphere with a star symbol. The closer the system is to the star, the higher the forward speed of the gait. Symmetric gaits are marked with a solid line, all symmetric gaits have $v_y = 0$. The figure shows that symmetric robust walking moves away from the apex of the hemisphere as energy increases, i.e. it becomes slower. At 830 J symmetric robust walking is constrained to the rightmost side of the viability region reducing the speed of this gait considerably. Furthermore, at this energy the region of symmetric walking breaks down into two unconnected segments. This is also evident in Fig. 3.1 where the maximum speed of symmetric robust walking shows a strong slowdown with a sudden change of slope. The

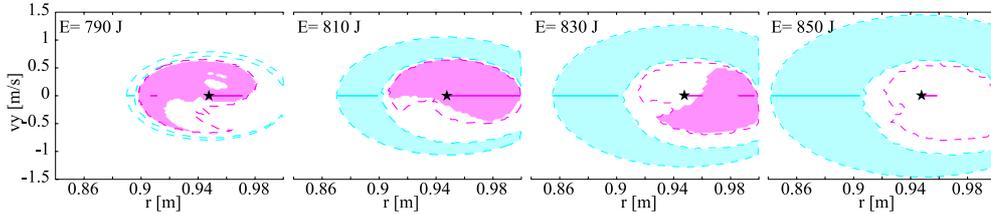


Figure 3.2: (Color online) Robust and viability regions. Areas inside dashed lines are viability regions. The colored areas show robust regions. The black star represents the apex of the hemisphere where, for the given energy, all possible states lie. The solid line shows the set of states corresponding to a periodic and symmetric gait. The (blue) light gray color represents \mathcal{R} and the (magenta) dark gray color represents \mathcal{W} .

latter is a consequence of the rupture of the symmetric gait region. This milestone in the evolution of the gait can be used as a natural trigger for a gait transition.

The evolution of the area of robust walking, $P^{\mathcal{W}}(2^\circ)$, and robust running, $P^{\mathcal{R}}(2^\circ)$, are shown in detail in Figure 3.3. This figure shows that, at low energy, robust walking covers a wide region of the viable states of the system, while at high energy robust running covers a wider area. Around 800 J both robust gaits have similar area. Based on robustness alone, this will imply a transition. However, symmetric robust walking intersects the apex of the hemisphere producing the fastest forward speed up to energies of 810 J, favoring walking in terms of energy efficiency. When the energy is increased further, the area of robust walking decreases and symmetric robust walking is constrained to low speeds. Due to these facts, at energies close to 840 J, the speed of symmetric robust walking and running match. For higher energies the gait transition is imminent, since the only robust gait remaining is symmetric running.

3.3.2 Mechanism of gait transitions

Assuming that during locomotion the fastest robust gait patterns are preferred over slower or non-robust ones, we see that for energies below 840 J walking is the gait of choice and for energies above that value running would be chosen. Therefore, we study viable transitions at 840 J and compare them with an experimental observation of human gait transition. We consider transitions only when all angles of attack used in the process can be chosen from an interval of length 2° or greater, i.e. we define admissible transitions using the concept of viability (sec. 3.2.2).

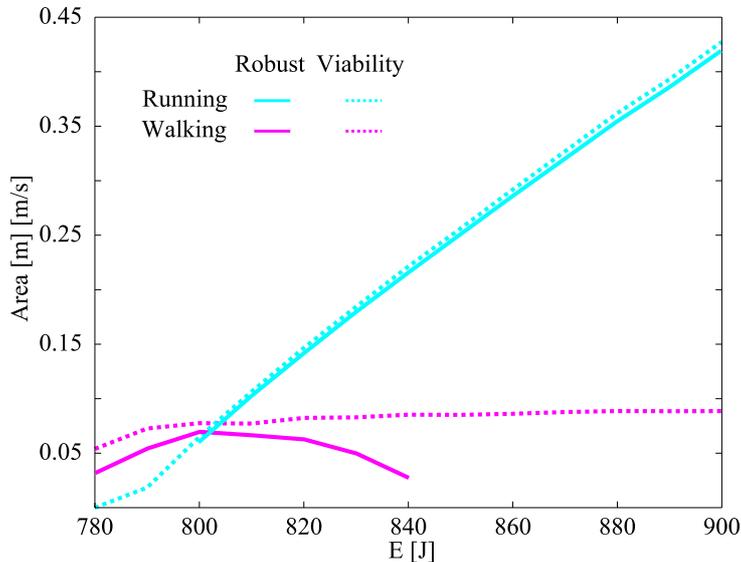


Figure 3.3: (Color online) Area of the viability regions and robust regions as a function of the energy. The dotted lines show the area of the viability region. The solid line shows the area of the robust regions. The (blue) light gray color represents \mathcal{R} and the (magenta) dark gray color represents \mathcal{W} .

We consider two mechanisms to execute gait transitions between symmetric robust gaits (symmetric gaits are known to be self-stable and therefore a good choice for stable locomotion, see [40]). The first mechanism, which can only be used from walking to running, consist in moving from the robust region of walking to the viability (non-robust) region of the same gait, and from there select an angle of attack to go to the robust region of running. This mechanism can be used in robust walking between 830 J and 840 J (see Figure 3.4). The second mechanism consist in going from a robust region of a given gait (walking or running) directly to the robust region of a different gait. This mechanism is applicable for robust running between 830 J and 840 J while in robust walking is only applicable around 840 J.

These mechanisms can be further constrained by selecting desired properties of the final gait. One possibility is to execute a transition in such a way that the final gait has the same (or as close as possible) Froude number as the initial gait. Another possibility is to execute a transition that sets the hip excursion of the new gait to a desired value (see Figure 3.5 for a graphical description). These constraints are referred in this study as strategies and they are used for the comparison between our simulated results and experimental data presented in the next section.

3.3.3 Qualitative Prediction of Biomechanical Observables

As we mention before, the biomechanical observables used to compare our results with experimental data are: Froude number, hip excursion, gait duty factor and vertical ground reaction forces. In the support material, we extended this comparison to include change of phase and angle of attack sequences. We compare all our simulations against the experimental data reported in Figure 2. of [51], we will refer to this data as “experimental data” or “the experiment”.

Figure 3.4 shows the transition regions at two energy levels. We painted the robust regions of running and walking with a solid color, the shaded regions inside these are transitions regions where the system can change the gait. The diagonal shading corresponds to regions where the system can change between robust gaits (non-symmetric) in only one step. The horizontal shading delimits the region where the system can go to the non-robust transition region, as described in 3.3.2. The right panel shows examples of a walking to running, and a running to walking transition using the two mechanisms mentioned earlier. The system starts at symmetric robust walking (1), in the first step it moves to the non-robust transition region (2*) and executes the transition to robust running (3*). With two further steps the system is able to reach symmetric robust running (4-5). The transition in the other direction starts at symmetric robust running (5). Then the system moves to the robust transition region (6*) from which, in a single step, it changes to robust walking (7*). With two more steps the system reaches symmetric robust walking (8-9). In both transitions, the hip excursion was kept as constant as possible.

Figure 3.5 shows the Froude number and the hip excursion of all symmetric robust gaits at 840 J. As indicated in the figure, vertical transitions keep the hip excursion constant, while horizontal transitions produce gaits with the same Froude number. Figure 3.6 shows time series of hip excursion and duty factor for a transition at constant hip excursion, together with a transition at constant Froude number. In both situations we obtain a Froude number that is about 60% smaller than the one found in human gait transitions which is around of 0.5 [51]. Nevertheless the SLIP model provides the best Froude number estimation to the date, when compared to other simple models, e.g. the IP model.

Ground reaction forces prior to the transition from walking to running have three main characteristics [52]. Firstly, they present an asymmetric double bell-shaped profile. Secondly, the earlier peak becomes bigger than the later one and, thirdly the depression between the peaks becomes more

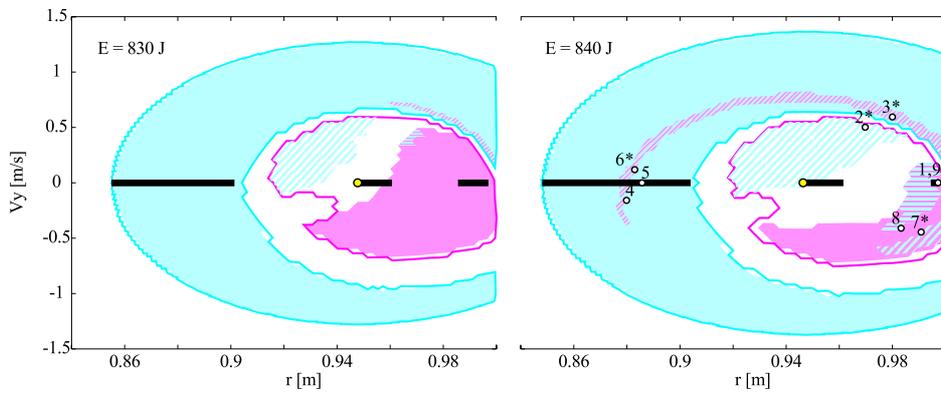


Figure 3.4: (Color online) Viable transitions at two energy levels. The viability regions are the area inside the solid lines. The colored areas show the robust regions. Shaded regions inside these are viable transitions regions. Diagonal shading corresponds to regions where the system can change between robust gaits (non-symmetric) in only one step. The horizontal shading delimits the region where the system can go to the non-robust transition region. The right panel shows an example of two transition using both mechanisms. The system goes from symmetric robust walking to symmetric robust running (1-5), the transition occurs between starred states. The transition in the other direction is indicated with steps (5-9).

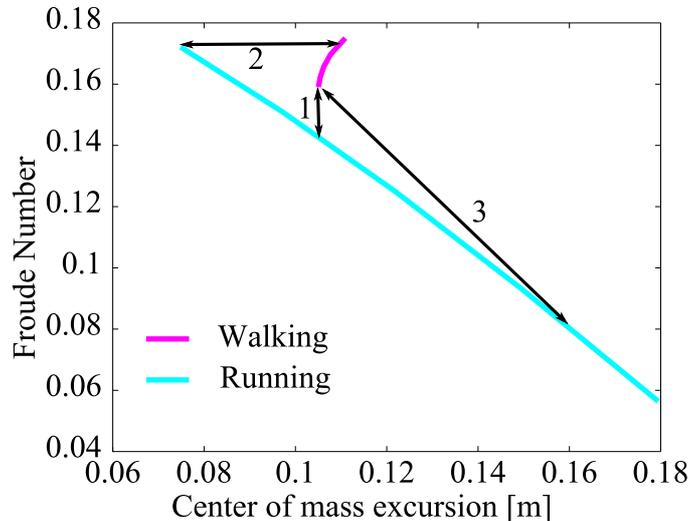


Figure 3.5: (Color online) Froude number versus hip excursion for symmetric robust running and walking. The (blue) light gray color represents \mathcal{R} and the (magenta) dark gray color represents \mathcal{W} . The arrows indicate the transitions study herein: constant hip excursion (1), constant Froude number (2) and (3) relative change of the amplitude of the hip excursion fitted to experimental data.

accentuated in the last step of walking, exactly before the transition. In the case of the transition from running to walking, it was reported that the vertical ground reaction forces decrease during the steps prior to the transition.

In Figure 3.7 we have plotted the vertical ground reaction forces for three different simulated examples. The first row of panels shows transitions from walking to running, and the second row of panels shows transitions in the other direction. Panels (a) and (b) show transitions keeping the Froude number constant. Panels (c) and (d) show transitions at constant hip excursion. The last example, presented in the panels (e) and (f), shows transitions that match the change in amplitude that was observed in the experiment. All cases qualitatively match the characteristics of the ground reactions reported in [52]. The decrement in the force of the last running step is due to the support of the second foot. A reduction of the peak in more than one step appears only on the case where we matched the hip excursion of the experimental data.

In Table 3.1, we present a summary of the comparison between the simulated examples and the experimental data. Each column is discussed below.

- Due to the variety of transitions that can be generated with the model,

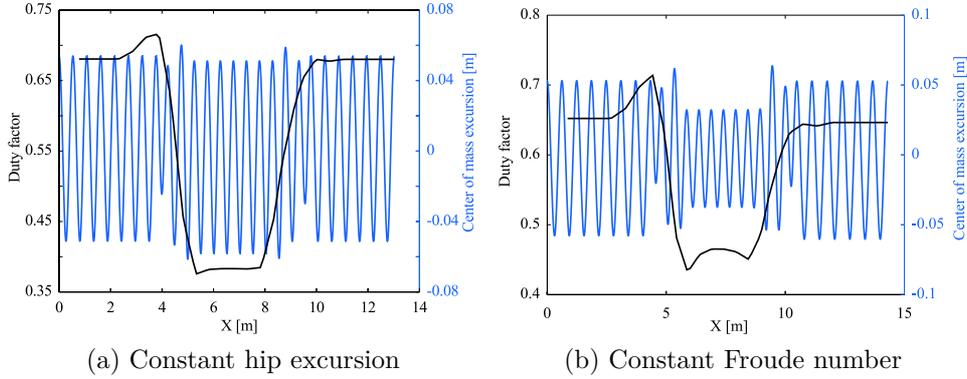


Figure 3.6: (Color online) Hip excursion and gait duty factor for transition at constant hip excursion (a); and constant Froude number (b). The (blue) light gray color represents the hip excursion and the black line represents the duty factor.

the number of steps to execute them can be select in a wide range, at least from 3 to 8 steps.

- From Figure 3.5 we can see that the Froude number of all these transitions are lower than 0.5, this reflects the fact that the simulations have lower forward speeds (v_x) than the observed in humans.
- As pointed before, the many transitions that can be simulated, permit the matching of the relative change in hip excursion (Δr) measured in the experiment.
- In all simulated transitions the vertical ground reaction forces (F_y) are qualitatively well reproduced.
- The selection of the angle of attack are qualitative similar to what we found in the experimental case: the system moves progressively from one gait to the other changing the angle of attack at each step. However, the oscillation of the hip before and after the simulated transitions presents a change of phase ($\Delta\phi$) that not always coincide with what is observed in reality. Details for these two observables are presented in the support material.

3.3.4 Robust Hopping Gait

At 840 J we identify a transition region in robust walking where the system can go in one step to robust running. Among the states in this transition

Strategy	# Steps	v_x	Δr	F_y	$\Delta\alpha$	$\Delta\phi$
Const. Froude number	✓	✗	✗	✓	✓	✗
Const. hip excursion	✓	✗	✗	✓	✓	✗
Fitting experiment	✓	✗	✓	✓	✓	✗

Table 3.1: Comparison between three transition strategies and experimental data. The symbol ✓ indicates qualitative matching between simulation and experiment, while the symbol ✗ indicates the opposite. v_x : forward speed of the center of mass; Δr : relative change in hip excursion before and after transition; F_y : vertical ground reaction forces; $\Delta\alpha$: change of the angle of attack during transition; $\Delta\phi$: change in phase of the oscillations of the hip before and after transition.

region, there are some that are mapped directly into the transition region of robust running. By selecting alternatively the right angles of attack, the system can sequentially walk and run, producing the hopping gait. Fig. 3.8 shows an example of this gait. By looking at the vertical ground reaction forces in the figure, we see the different phases that compose this gait; from single stance phase to double stance phase then to single stance phase and finally to flight phase.

3.4 Discussion

Herein we have modeled bipedal locomotion using the SLIP model. This model conserves the total mechanical energy and at first glance it may seem inapposite for the prediction of gait transitions, since work has to be done on the system to increase the speed of locomotion. Nevertheless, by looking at the behavior of the model at different energies, we can emulate the situation where work is done on the system.

We proposed robustness as a new measure of the easiness of locomotion. Robustness measures the level of attention that needs to be dedicated to take a step; the more robust a gait is, the less attention that is needed to take the next step.

According to our results, the selection of the gait can be based on two criteria: efficiency, which is the selection of the gait with the highest forward speed; and robustness, which defines how easy is to maintain the given gait. This second criterion is consistent with the experimental results of attentional demand in locomotion reported in [50]. Based on these criteria, walking is the best choice for energies below 840J, and running is more appropriate for higher energies. This resembles what is observed in human locomotion.

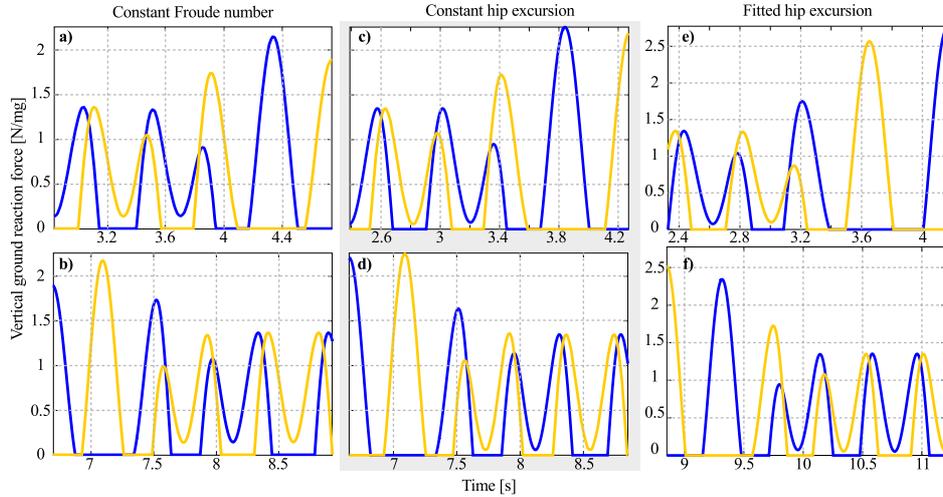


Figure 3.7: Vertical ground reaction forces during transitions. The six panels show a transition from symmetric robust walking to symmetric robust running with three different strategies, (a)-(b) constant Froude number, (c)-(d) constant hip excursion, (e)-(f) hip excursion similar to the experimental data. The forces present an asymmetric double bell-shaped profile. In the walking to running transition, (a)-(c) and (e), the earlier peak becomes bigger than the later one, exactly before the transition. The transitions in the other direction, running to walking (b)-(d) and (f) show vertical ground reaction forces that decrease considerably in the last running step due to the support of the second foot. The selection of a hip excursion similar to the experimental data introduces a progressive reduction of the force peak in more than one step (f). All forces are normalized with respect to the weight of the system.

Using robustness as the leading criterion, we identify transition regions that allow the system to go from one gait to the other even in the case of imprecise angle selection. These transition regions are present for energies from 830 J to 840 J (Fig. 3.4). At 840 J, symmetric robust running and walking share all the possible velocities, facilitating gait transitions. In the case of an increment of energy, to keep robustness and move forward faster, a walking system can execute a transition to robust running at 840 J. The transition can be reversed when the system decreases its energy. Note that the mechanisms of transition shown in Fig. 3.4 (right panel), have the following properties. One mechanism connects the robust region of both gaits, while the other one connects the non-robust viability region of walking with robust running. The latter mechanism is not reversible, meaning that the system cannot go from running back to this region in a single step. The

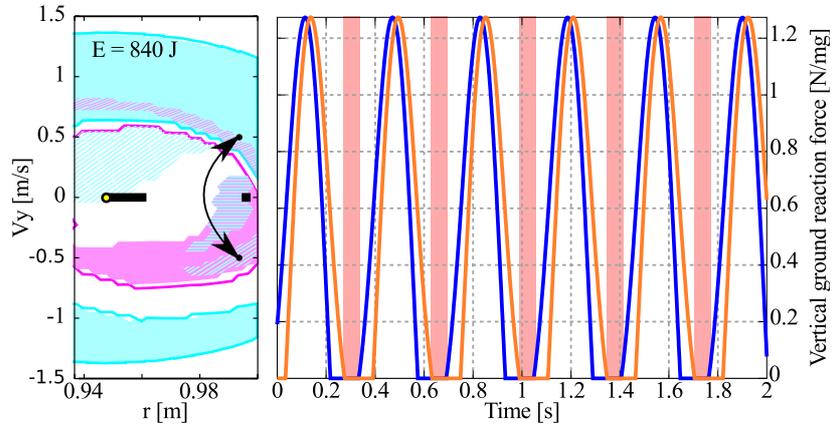


Figure 3.8: (Color online) Vertical ground reaction forces during hopping. Panel (a) shows the transition regions in section \mathcal{S} for $E = 840$ J; the arrows show the states in the robust transition region that are used alternately. Panel (b) shows the ground reaction forces for each leg. The (pink) gray rectangles show the different flight phases. The forces from the legs are indicated with solid lines with different colors.

transitions connecting robust regions are reversible and the system can oscillate between the two gaits robustly. Is in this situation where the hopping gait emerges. This locomotion pattern is frequently used by children when playing joyfully.

The existence of non-empty transition regions (Fig. 3.5) implies that the system has multiple alternatives to change gaits. These alternatives will produce different changes of forward speed and hip excursion. We show three different scenarios: constant hip excursion, hip excursion similar to experimental data and constant Froude number.

When the transition matches the hip excursion of the experimental data, the Froude number varies from 0.16 in walking to 0.08 in running, while in the experiment it is almost constant (slowly varying treadmill speed, see [51] for details on the experiment). As explained before, in all simulated cases the absolute values of Froude number are lower than in the experiments. The hip excursion has an amplitude of 5.2 cm in walking and 8.3 cm which also similar to the one reported in [51] which is around 7 cm.

When the transition keeps the Froude number constant the hip excursion decreases from 5.7 cm in walking to 3.7 cm in running. This contradicts the behavior observed in our experimental data. The simulated Froude number for this transition is about 0.17.

The robustness criterion induces an underestimation of the forward speed

at gait transitions. The highest Froude number archived using the previous strategies is around one third of the one observed in humans (0.5). However, given the strong simplifications in the model the result is encouraging. To reduce the gap between simulated and experimental Froude number, the model can be extended to include the displacement of the point where the leg is in contact with the ground during the stance phase [53].

All transitions presented here produce similar results concerning the duty factor. Walking has a duty factor around 0.7 and running has a duty factor around 0.4, in accordance with the experiment. Furthermore, in all transitions from walking to running the model predicts a progressive change in the vertical component of the reaction forces, i.e. the relation between the first and the second peak of the force during the transition. This also applies to the transitions from running to walking. In particular, the ground reaction forces corresponding to transitions matching the hip excursion of the experimental data (Fig. 3.7) introduces a progressive reduction of the force peak in more than one step. All these results qualitatively reproduce the experimental results reported in [52].

3.5 Conclusion

The comparison between experimental data and the simulation using the SLIP model shows that the model is not able to generate accurate quantitative predictions. Most strikingly, the forward speed in the simulations are considerable slower than that observed experimentally. This difficulty can be overcome by adding a more detailed description of the contact between leg and ground. Nevertheless, the SLIP model can be used as a conceptual model to explain the many aspects of bipedal locomotion such as the mechanics of running, walking, hopping and gait transitions.

Our findings indicate that robustness can play an important role in inducing gait transition, complementing to the usual view focused solely in energy expenditure. The robustness criterion is analogous to the attentional demand during locomotion and may play an important role deciding the gait transition events. To our knowledge this is the first time such a criterion is included in a numerical model of locomotion.

Acknowledgements The research leading to these results has received funding from the European Community's Seventh Framework Programme FP7/2007-2013-Challenge 2-Cognitive Systems, Interaction, Robotics- under grant agreement No 248311-AMARSi.

Authors contribution **HMS** developed the computational and mathematical model, run the simulations and performed data analysis. **JPC** collaborated in development of the mathematical models, the data analysis and the interpretation of results. **YI** collected and contributed the experimental data. All authors contributed to the writing of this manuscript.

Chapter 4

Synthesis and Adaptation of Effective Motor Synergies for the Solution of Reaching Tasks

4.1 Introduction

Humans are able to perform a wide variety of tasks with great flexibility; learning new motions is relatively easy, and adapting to new situations (e.g. change in the environment or body growth) is usually dealt with no particular effort. The strategies adopted by the central nervous system (CNS) to master the complexity of the musculoskeletal apparatus and provide such performance are still not clear. However, it has been speculated that an underlying modular organization of the CNS may simplify the control and provide the observed adaptability. There is evidence that the muscle activity necessary to perform various tasks (e.g. running, walking, keeping balance, reaching and other combined movements) may emerge from the combination of predefined muscle patterns, the so-called *muscle synergies* [54]. This organization seems to explain muscle activity across a wide range of combined movements [55–57].

The scheme of muscle synergies is inherently flexible and adaptable. Different actions are encoded by specific combinations of a small number of predefined synergies; this reduces the computational effort and the time required to learn new useful behaviors. The learning scheme can be regarded as developmental since information previously acquired (i.e. synergies) can be reused to generate new behaviors[58]. Finally, improved performance can be easily achieved by introducing additional synergies. Thus, the hypothetical scheme of muscle synergies would contribute to the autonomy and the

flexibility observed in biological systems, and it could inspire new methods to endow artificial agents with such desirable features.

In this paper we propose a method to control a dynamical system (i.e. the agent) in point-to-point reaching tasks by linear combinations of a small set of predefined actuations (i.e. synergies). Our method initially solves the task in state variables by interpolation; then, it identifies the combination of synergies (i.e. actuation) that generate the closest kinematic trajectory to the computed interpolant. Additionally, we propose a strategy to synthesize a small set of synergies that is tailored to the task and the agent. The overall method can be interpreted in a developmental fashion; i.e. it allows the agent to autonomously synthesize and update its own synergies to increase the performance of new reaching tasks.

Other researchers in robotics and control engineering have recently proposed architectures inspired by the concept of muscle synergies. In [59] the authors derive an analytical form of a set of primitives that can drive a feedback linearized system (known analytically) to any point of its configuration space. In [60] the authors present a numerical method to identify synergies that optimally drive the system over a set of desired trajectories. This method does not require an analytical description of the system, and it has the advantage of assessing the quality of the synergies in task space. However, it is computationally expensive as it involves heavy optimizations. In [61] muscle synergies are identified by applying an unsupervised learning procedure to a collection of sensory-motor data obtained by actuating a robot with random signals. In [62] the architecture of the dynamic movement primitives (DMP) is proposed as a novel tool to formalize control policies in terms of predefined differential equations. Linear combinations of Gaussian functions are used as inputs to modify the attractor landscapes of these equations, and to obtain the desired control policy.

In contrast to these works, our method to synthesize synergies does not rely on feedback linearization, nor on repeated integrations of the dynamical system. The method is grounded on the input-output relation of the dynamical system (as in [61]), and it provides a computationally fast method to obtain the synergy combinatorics to solve a given task. Furthermore, our method is inherently adaptable as it allows the on-line modification of the set of synergies to accommodate to new reaching tasks.

4.2 Definitions and Methods

In this section we introduce the mathematical details of the method we propose. After some definitions, we present the core element of our method: a

general procedure to compute actuations that solve point-to-point reaching tasks (see Sec. 4.2.1). Subsequently, in Section 4.2.2, we propose a framework for the synthesis and the development of a set of synergies.

Let us consider a differential equation modeling a physical system $\mathcal{D}(\mathbf{q}(t)) = \mathbf{u}(t)$, where $\mathbf{q}(t)$ represents the time-evolution of its configuration variables (their derivatives with respect to time are $\dot{\mathbf{q}}(t)$), and $\mathbf{u}(t)$ is the actuation applied. Inspired by the hypothesis of muscle synergies¹ [54], we formulate the actuation as a linear combination of predefined motor co-activation patterns:

$$\mathbf{u}(t) = \sum_{i=1}^{N_\phi} \phi_i(t) b_i := \mathbf{\Phi}(t) \mathbf{b}, \quad (4.1)$$

where the functions $\phi_i(t) \in \mathbf{\Phi}$ are called *motor synergies*. The notation $\mathbf{\Phi}(t)$ describes a formal matrix where each column is a different synergy. If we consider a time discretization, $\mathbf{\Phi}(t)$ becomes a $N \dim(\mathbf{q})$ -by- N_ϕ matrix, where N is the number of time steps, $\dim(\mathbf{q})$ the dimension of the configuration space and N_ϕ the number of synergies.

We define *dynamic responses* (DR) of the set of synergies as the responses $\boldsymbol{\theta}_i(t) \in \boldsymbol{\Theta}$ of the system to each synergy (i.e. forward dynamics):

$$\mathcal{D}(\boldsymbol{\theta}_i(t)) = \phi_i(t) \quad i = 1 \dots N_\phi. \quad (4.2)$$

with initial conditions chosen arbitrarily.

4.2.1 Solution to point-to-point reaching tasks

A general point-to-point reaching task consists in reaching a final state $(\mathbf{q}_T, \dot{\mathbf{q}}_T)$ from an initial state $(\mathbf{q}_0, \dot{\mathbf{q}}_0)$ in a given amount of time T :

$$\begin{aligned} \mathbf{q}(0) &= \mathbf{q}_0, & \dot{\mathbf{q}}(0) &= \dot{\mathbf{q}}_0, \\ \mathbf{q}(T) &= \mathbf{q}_T, & \dot{\mathbf{q}}(T) &= \dot{\mathbf{q}}_T. \end{aligned} \quad (4.3)$$

Controlling a system to perform such tasks amounts to finding the actuation $\mathbf{u}(t)$ that fulfills the point constraints² (4.3). Specifically, assuming that the synergies are known, the goal is to identify the appropriate synergy combiners \mathbf{b} . In this paper we consider only the subclass of reaching tasks that impose motionless initial and final postures, i.e. $\dot{\mathbf{q}}_T = \dot{\mathbf{q}}_0 = 0$.

The procedure consists of, first, solving the problem in kinematic space (i.e. finding the appropriate $\mathbf{q}(t)$), and then computing the corresponding

¹With respect to the model of time-varying synergies, in this paper we neglect the synergy onset times.

²In this paper we assume that the initial conditions of the systems are equal to $(\mathbf{q}_0, \dot{\mathbf{q}}_0)$

actuations. From the kinematic point of view, the task can be seen as an interpolation problem; i.e. $\mathbf{q}(t)$ is a function that interpolates the data in (4.3). Therefore, a set of functions is used to build the interpolant trajectory that satisfy the constraints imposed by the task; these functions are herein the dynamic responses of the synergies:

$$\mathbf{q}(t) = \sum_{i=1}^{N_\theta} \boldsymbol{\theta}_i(t) \mathbf{a}_i := \boldsymbol{\Theta}(t) \mathbf{a}, \quad (4.4)$$

where the vector of combinators \mathbf{a} is chosen such that the task is solved. As mentioned earlier, if time is discretized, $\boldsymbol{\Theta}(t)$ becomes a $N \dim(\mathbf{q})$ -by- N_θ matrix, where N_θ is the number of dynamic responses. The quality of the DR as interpolants is evaluated in sections 4.3.

Once a kinematic solution has been found (as linear combination of DRs), the corresponding actuation can be obtained by applying the differential operator; i.e. $\mathcal{D}(\boldsymbol{\Theta}(t) \mathbf{a}) = \tilde{\mathbf{u}}(t)$. Finally, the vector \mathbf{b} can be computed by projecting $\tilde{\mathbf{u}}(t)$ onto the synergy set $\boldsymbol{\Phi}$. If $\tilde{\mathbf{u}}(t)$ does not belong to the linear span of $\boldsymbol{\Phi}$, the solution can only be approximated in terms of a defined norm (e.g. Euclidean):

$$\mathbf{b} = \arg \min_{\mathbf{b}} \|\tilde{\mathbf{u}}(t) - \boldsymbol{\Phi}(t) \mathbf{b}\|. \quad (4.5)$$

When the time is discretized, all functions of time becomes vectors and this equation can be solved explicitly using the psuedoinverse of the matrix $\boldsymbol{\Phi}$,

$$\boldsymbol{\Phi}^+ \tilde{\mathbf{u}} = \boldsymbol{\Phi}^+ \mathcal{D}(\boldsymbol{\Theta} \mathbf{a}) = \mathbf{b}. \quad (4.6)$$

This equation highlights the operator $\boldsymbol{\Phi}^+ \circ \mathcal{D} \circ \boldsymbol{\Theta}$ (\circ denotes operator composition) as the mapping between the kinematic combinators \mathbf{a} (kinematic solution) and the synergy combinators \mathbf{b} (dynamic solution). Generically, this operator represents a nonlinear mapping $\mathcal{M} : \mathbb{R}^{N_\theta} \rightarrow \mathbb{R}^{N_\phi}$, and it will be discussed in Section 4.4.

To assess the quality of the solution we define the following measures:

Interpolation error: Measures the quality of the interpolant $\boldsymbol{\Theta}(t) \mathbf{a}$ with respect to the task. Strictly speaking, only the case of negligible errors corresponds to interpolation. A non-zero error indicates that the trajectory $\boldsymbol{\Theta}(t) \mathbf{a}$ only approximates the task

$$\text{err}_I = \sqrt{\|\mathbf{q}_T - \boldsymbol{\Theta}(T) \mathbf{a}\|^2 + \|\dot{\boldsymbol{\Theta}}(T) \mathbf{a}\|^2}, \quad (4.7)$$

where $\|\cdot\|$ denotes the Euclidean norm, and the difference between angles are mapped to the interval $(-\pi, \pi]$.

Projection error: Measures the distance between the actuation that solves

the task $\tilde{\mathbf{u}}(t)$, and the linear span of the synergy set Φ

$$\text{err}_P = \sqrt{\int_0^T \|\tilde{\mathbf{u}}(t) - \Phi(t)\mathbf{b}\|^2 dt}. \quad (4.8)$$

Forward dynamics error: Measures the error of a trajectory $\tilde{\mathbf{q}}(t, \boldsymbol{\lambda})$ generated by an actuation $\Phi(t)\boldsymbol{\lambda}$, in relation to the task.

$$\text{err}_F = \sqrt{\|\tilde{\mathbf{q}}(T, \boldsymbol{\lambda}) - \mathbf{q}_T\|^2 + \|\dot{\tilde{\mathbf{q}}}(T, \boldsymbol{\lambda}) - \dot{\mathbf{q}}_T\|^2}. \quad (4.9)$$

Replacing $\tilde{\mathbf{q}}(t, \boldsymbol{\lambda})$, \mathbf{q}_T and $\dot{\mathbf{q}}_T$ with their corresponding end-effector values provides the forward dynamics error of the end-effector.

4.2.2 Synthesis and Development of Synergies

The synthesis of synergies is carried on in two phases: exploration and reduction. The exploration phase consists in actuating the system with an extensive set of motor signals Φ_0 in order to obtain the corresponding DRs Θ_0 . The reduction phase consists in solving a small number of point-to-point reaching tasks in kinematic space (that we call *proto-tasks*) by creating the interpolants using the elements of set Θ_0 , as described in Eq. (4.4). These solutions are then taken as the elements of the reduced set Θ . Finally, the synergy set Φ is computed using relation (4.2), i.e. inverse dynamics. As a result, there will be as many synergies as the number of the proto-tasks (i.e. $N_\phi = N_\theta$). The intuition behind this reduction is that the synergies that solve the proto-tasks may capture essential features both of the task and of the dynamics of the system. Despite the non-linearities of \mathcal{D} , linear combination of these synergies might be useful to solve point-to-point reaching tasks that are similar (in terms of Eq. (4.3)) to the proto-tasks (see Sec. 4.3).

The number of proto-tasks as well as their specific instances determine the quality of the synergy-based controller. To obtain good performance in a wide variety of point-to-point reaching tasks, the proto-tasks should cover relevant regions of the state space (see Sec. 4.3). Clearly, the higher the number of different proto-tasks, the more regions that can be reached with good performance. However, a large number of proto-tasks (and the corresponding synergies) increases the dimensionality of the controller. In order to tackle this trade-off, we propose a procedure that parsimoniously adds a new proto-task only when and where it is needed: if the performance in a new reaching task is not satisfactory, we add a new proto-task in one of the regions with highest projection error or we modify existing ones.

4.3 Results

We apply the methodology described in Section 4.2 to a simulated planar kinematic chain (see [63] for model details) modeling a human arm[64]. In the exploration phase, we employ an extensive set of motor signals Φ_0 to actuate the arm model and generate the corresponding dynamic responses Θ_0 . The panels in the first row of Fig. 4.1 show the end-effector trajectories resulting from the exploration phase. We test two different classes of motor signals: actuations that generate minimum jerk end-effector trajectories (100 signals), and low-passed uniformly random signals (90 signals). In order to evaluate the validity of the general method described in Sec. 4.2.1, we use the sets Φ_0 and Θ_0 to solve 13 different reaching tasks without performing the reduction phase. The second row of Fig. 4.1 depicts the trajectories drawn by the end-effector when the computed mixture of synergies are applied as actuations (i.e. forward dynamics of the solution). It has to be noted how the nature of the solutions (as well as that of the responses), depends on the class of actuations used. The maximum errors are reported in Table 4.1. The results are highly satisfactory for both the classes of actuations, and show the validity of the method proposed. Since the reduction phase has not been performed, the dimension of the combinator vectors \mathbf{a} and \mathbf{b} equals the number of actuations used in the exploration.

	Min. Jerk	Random
err _I	10 ⁻¹⁵	10 ⁻¹⁵
err _P	10 ⁻⁵	10 ⁻³
err _F	10 ⁻⁴	10 ⁻³

Table 4.1: Order of the maximum errors obtained by using Φ_0 and Θ_0 (no reduction phase).

The objective of the reduction phase is to generate a small set of synergies and DRs that can solve desired reaching tasks effectively. As described in Section 4.2.2, this is done by solving a handful of proto-tasks. The number (and the instances) of these proto-tasks determines the quality of the controller. Figure 4.2 shows the projection error as a function of the number of proto-tasks. The reduction is applied to the low-passed random signal set. Initially, two targets are chosen randomly (top left panel); subsequent targets are then added on the regions characterized by higher projection error. As it can be seen, the introduction of new proto-tasks leads to better performance on wider regions of the end-effector space, and eventually the whole space can be reached with reasonable errors. In fact, the figure shows that this

procedure decreases the average projection error to 10^{-3} (comparable to the performance of the whole set Φ_0 , see Tab. 4.1) and reduces the dimension of the combinator vector to 6, a fifteen-fold reduction. This result shows that a set of “good” synergies can drastically reduce the dimensionality of the controller, while maintaining similar performance. The bottom right panel of the figure shows the forward dynamics error of the end-effector obtained with the 6 proto-tasks. Comparing this panel with the bottom left one, it can be seen that the forward dynamics error of the end-effector reproduces the distribution of the projection error, rendering the latter a good estimate for task performance.

To further demonstrate that the reduction phase we propose is not trivial, we compare the errors resulting from the set of 6 synthesized synergies, with the errors corresponding to 100 random subsets of size 6 drawn from the set of low-passed random motor signals. Figure 4.3 shows this comparison. The task consists in reaching the 13 targets in Fig. 4.1. The boxplots correspond to the errors of the random subsets, and the filled circles to the errors of the synergies resulting from the reduction phase. Observe that, the order of the error of the reduced set is, in the worst case, equal to error of the best random subset. However, the mean error of the reduced set is about 2 orders of magnitude lower. Therefore, the reduction by proto-tasks can produce a parsimonious set of synergies out of a extensive set of actuations. Evaluating the performance with different classes of proto-tasks (e.g. catching, hitting, via-points) is postponed to future works.

4.4 Discussion

The results shown in the previous section justify the interpretation of the methodology as a developmental framework. Initially, the agent explores its sensory-motor system employing a variety of actuations. Later, it attempts to solve the first reaching tasks (proto-tasks), perhaps obtaining weak performance as the exploration phase may not have produced enough responses yet (see the box-plots in Fig. 4.3). If the agent finds an acceptable solution to a proto-task, it is used to generate a new synergy (populating the set Φ), otherwise it continues with the exploration. The failure to solve tasks of importance for its survival, could motivate the agent to include additional proto-tasks; Figure 4.2 illustrates this mechanism. As it can be seen, the development of the synergy set incrementally improves the ability of the agent to perform point-to-point reaching. Alternatively, existing proto-tasks could be modified by means of a gradient descent or other learning algorithms. In a nutshell, the methodology we propose endows the agent with the abil-

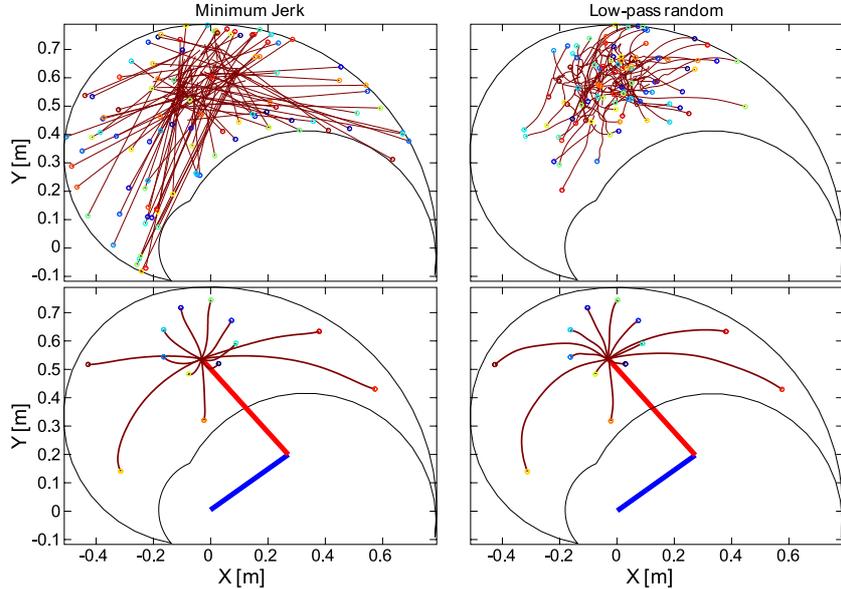


Figure 4.1: Comparison of explorations with two different classes of actuation: minimum jerk and low-passed random signal. Each panel shows the kinematic chain in its initial posture (straight segments). The limits of the end-effector are shown as the boundary in solid line.

ity to autonomously generate and update a set of synergies (and dynamic responses) that solve reaching tasks effectively.

Despite the difficulty of the mathematical problem (i.e. nonlinear differential operator), our method seems to generate a small set of synergies that span the space of actuations required to solve reaching tasks. This is not a trivial result, since these synergies over-perform many other sets of synergies randomly taken from the set Φ_0 (see Fig. 4.3). It appears as if the reduction phase builds features upon the exploration phase, that are necessary to solve new reaching tasks. To verify whether solving proto-tasks plays a fundamental role, our synergies could be compared with the principal components extracted from the exploration set. This verification goes beyond the scope of this paper.

An important aspect of our method is the relation between Θ and Φ (see Eq. (4.2)). This mapping makes explicit use of the body parameters (embedded in the differential operator \mathcal{D}), hence the synergies obtained can always be realized as actuations. The same cannot be said, in general, for synergies identified from numerical analyses of biomechanical data. Though some studies have verified the feasibility of extracted synergies as actuations [65], biomechanical constraints are not explicitly included in the extraction

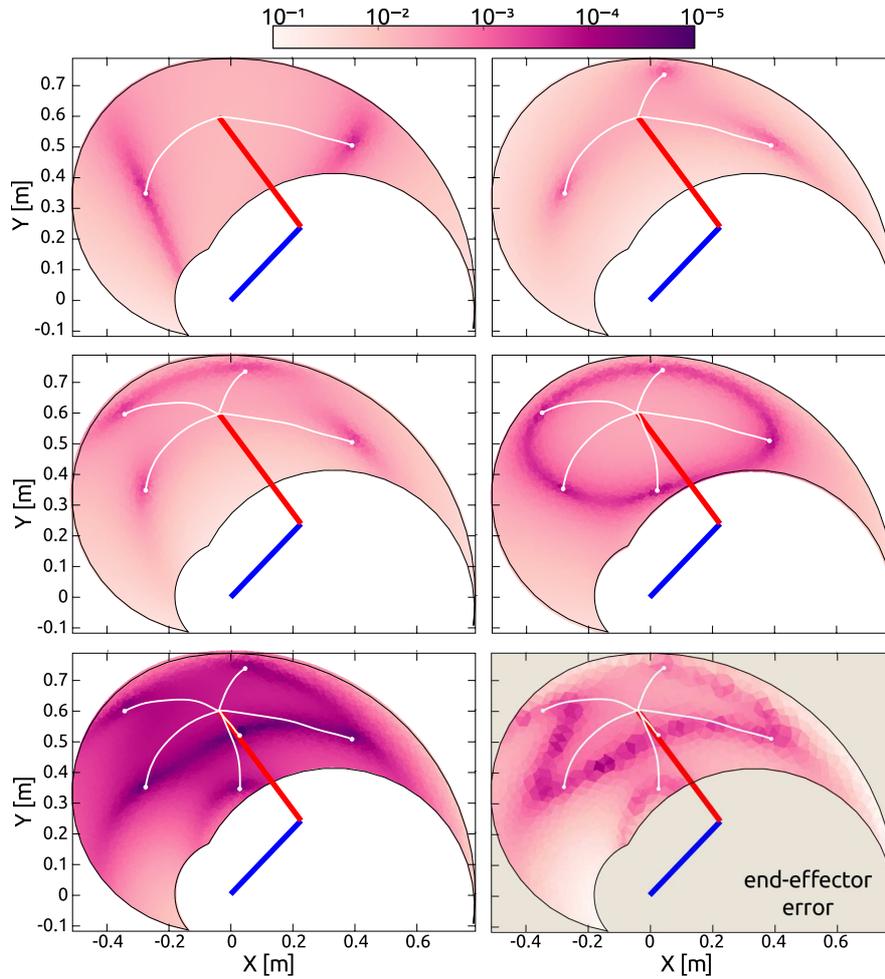


Figure 4.2: Selection of targets based on projection error. Each panel shows the kinematic chain in its initial posture (straight segments). The limits of the end-effector are the boundary of the colored regions. The color of each point indicates the projection error produced to reach a target in that position. The bottom right diagram shows the forward dynamics error of the end-effector using 6 proto-tasks (6 synergies).

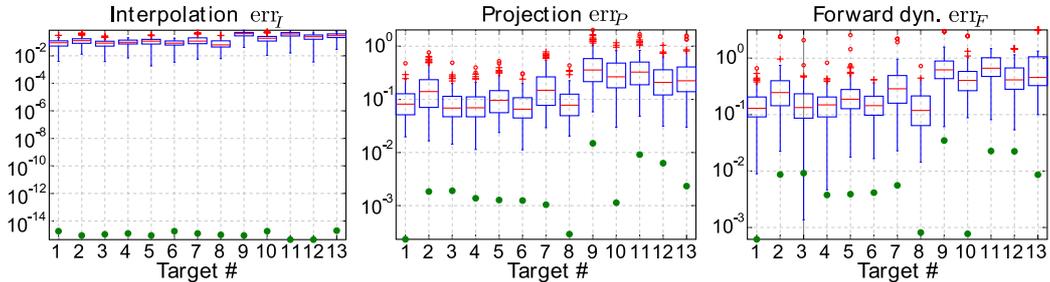


Figure 4.3: Evaluation of the reduction phase. Errors produced by subsets randomly selected from the exploration-actuations (boxplots) are compared with the errors obtained after the reduction phase (filled circles).

algorithms. Additionally, Eq. (4.2) provides an automatic way to cope with smooth variations of the morphology of the agent. That is, both the synergies and their dynamic responses evolve together with the body. In line with [59, 60], these observations highlight the importance of the body in the hypothetical modularization of the CNS.

Once the task is solved in kinematic space, the corresponding actuation can be computed using the explicit inverse dynamical model of the system (i.e. the differential operator \mathcal{D}). It might appear that there is no particular advantage in projecting this solution onto the synergy set. However, the differential operator might be unknown. In this case, a synergy-based controller would allow to compute the appropriate actuation by evaluating the mapping \mathcal{M} on the vector \mathbf{a} , hence obtaining the synergy combinator \mathbf{b} . Since \mathcal{M} is a mapping between two finite low-dimensional vector spaces, estimating this map may turn to be easier than estimating the differential operator \mathcal{D} . Furthermore, we believe that the explicit use of \mathcal{D} may harm the biological plausibility of our method. In order to estimate the map \mathcal{M} , the input-output data generated during the exploration phase (i.e. Φ_0 and Θ_0) could be used as learning data-set. Further work is required to test these ideas. Additionally, preliminary theoretical considerations (not reported here) indicate that the synthesis of synergies without the explicit knowledge of \mathcal{D} is also feasible.

Finally, the current formulation of the method does not include joint limits explicitly. The interpolated trajectories are valid, i.e. they do not go beyond the limits, due to the lack of intricacy of the boundaries. In higher dimensions, especially when configuration space and end-effector are not mapped one-to-one, this may not be the case anymore. Nevertheless, joint limits can be included by reformulating the interpolation as a constrained minimization problem. Another solution might be the creation of

proto-tasks with a tree-topology, relating our method to tree based path planning algorithms[66].

4.5 Conclusion and Future Work

The current work introduces a simple framework for the generation of open loop controllers based on synergies. The framework is applied to a planar kinematic chain to solve point-to-point reaching tasks. Synergies synthesized during the reduction phase over-perform hundreds of arbitrary choices of basic controllers taken from the exploration motor signals. Furthermore, our results confirm that the introduction of new synergies increases the performance of reaching tasks. Overall, this shows that our method is able to generate effective synergies, greatly reducing the dimensionality of the problem, while keeping a good performance level. Additionally, the methodology offers a developmental interpretation of the emergence of task-related synergies that could be validated experimentally.

Due to the nonlinear nature of the operator \mathcal{D} , the theoretical grounding of the method poses a difficult challenge, and it is the focus of our current research. Another interesting line of investigation is the validation of our method against biological data, paving the way towards a predictive model for the hypothesis of muscle synergies. Similarly, the development of an automatic estimation process for the mapping \mathcal{M} would further increase the biological plausibility of the model.

The inclusion of joint limits into the current formulation must be prioritized. Solving this problem will allow to test the method on higher dimensional redundant systems. Tree-based path planning algorithms may offer a computationally effective way to approach the issue.

Funds: The research leading to these results has received funding from the European Community’s Seventh Framework Programme FP7/2007-2013-Challenge 2 - Cognitive Systems, Interaction, Robotics- under grant agreement No 248311-AMARSi, and from the EU project RobotDoC under 235065 from the 7th Framework Programme (Marie Curie Action ITN).

Authors Contribution: **CA** and **JPC** worked on the implementation of the algorithm and the generation of the results reported here. The method was born during **JPC**’s visit to **AD**’s laboratory. **AD** provided material support for this development and uncountable conceptual inputs. All three authors have contributed to the creation of the manuscript. The authors list follows an alphabetical order.

Chapter 5

Locomotion without a brain: Physical Reservoir Computing in Tensegrity structures

5.1 Introduction

Embodiment has led to a revolution in robotics and it encompasses much of the research on the nature of cognition [67]. The idea of embodiment has many definitions, but they share a common notion. By not thinking of the agent body and its controller as two separate units, but instead taking the interaction of the body with its environment into account, a more attuned sensory representation is being generated. This in turn makes the task of complex control of locomotion easier. Indeed: the principle of embodiment implies amongst others that the direct physical interaction between the body and its environment is crucial for advanced cognitive processing by the agent. In this work, we will use a pragmatic viewpoint of embodiment and investigate the idea in particular by studying the computational powers of body dynamics.

Pfeifer and Bongard demonstrated the importance of embodiment in their book “How the body shapes the way we think”. In the first chapter of their book, they write [68] p. 19:

First, embodiment is an enabler for cognition or thinking: in other words, it is a prerequisite for any kind of intelligence. So, the body is not something troublesome that is simply there to carry the brain around, but it is necessary for cognition.

Indeed, the body is not just the brain’s interface to the world, but it is the combination of body and brain that defines an agent. Clark wrote in the introduction of his book “Being there: Putting brain, body and world together again” [69], which was largely influenced by Brooks’ work on embodied robotics [70, 71]:

We ignored the fact that the biological brain is, first and foremost, an organ for controlling the biological body. Minds make motions, and they must make them fast - before the predator catches you, or before your prey gets away from you. Minds are not disembodied logical reasoning devices.

This work, published 10 years before Pfeifer’s, in our opinion still tends to put the emphasis on a brain, but acknowledges that the body is an important factor in what defines an agent.

Although previous work on embodiment has demonstrated the importance of thinking about the agent as a unit of body and brain, it is unclear what the body is contributing in a computational sense.

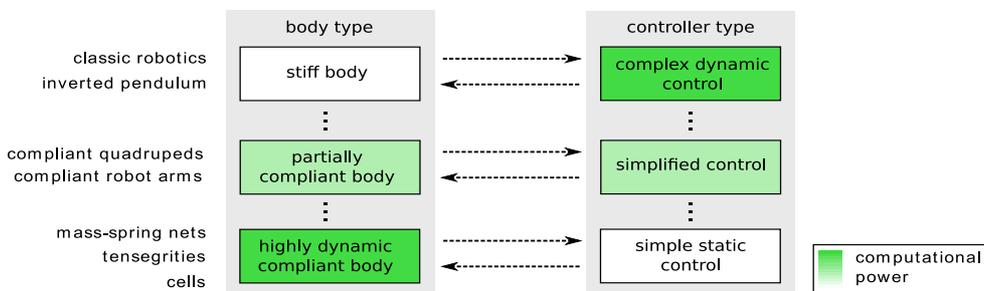


Figure 5.1: The computational tradeoff. Compliance and underactuation provide additional freedom to the system, which in turn corresponds to computational power which can be used to simplify the control problem.

In this work, we will study an exemplar family of highly dynamic body structures, controlled by the simplest possible “brains”. While this could be seen as an example of an extreme form of Morphological Computation [72], we propose to interpret the results presented here more broadly, namely as a particular implementation of the general principle of computation with complex non-linear dynamical systems with relatively simple adaptive controls, called Physical Reservoir Computing (PRC).

By doing so, we clearly demonstrate that it is possible on the one end to have very static bodies and highly complex controllers (as in classic robotics), while on the other end of the spectrum we can have a highly dynamic body, being controlled by a very simple controller. Still, both can perform similar “computations” in the environment. By investigating this extreme, we clearly demonstrate the spectrum of choices in how control can be implemented in the body-brain composite. Fig. 5.1 gives an overview of this spectrum. In this trade-off between brain and “body” computation, there are many known intermediate results: dead fish are propelled forward in a vortex [73, 74], single celled organisms (e.g. amoebae with pseudopodia) can be thought of as only computing using their “body” and chemical pathways as they do not possess neural substrates [75, 76], nematodes have rich

motor patterns with only a few neurons (302 for the hermaphrodite *C. elegans*) and thus their locomotion largely depends on the shape of their body [77, 78], the finger-tendon network in humans is responsible for a large part of the computational load required in fine control [79], it was demonstrated that the locomotion pattern of decerebrated cats can autonomously tune itself to the body-environment interaction [80–82], ...

Note that there are also important philosophical questions on the relationship between brain, body and the environment (see e.g. [83, 84]). The viewpoint that these three are separate, albeit interacting, entities has been challenged by the notion of embodied cognition: cognition is no longer seen as an exclusive property of the brain and the functionalities traditionally exclusively attributed to the brain, such as memory or complex transformations of sensory inputs, seem to be performed by other parts of the body as well. As such, the distinction between brain, body and environment becomes blurry, and the classical modular view on the relation between these three slowly disappears.

Recently, Hauser et al. [85, 86] showed that spring-mass nets have universal computational power¹, providing a theoretical foundation for Morphological Computation which in this setting is an instantiation of the field known as Reservoir Computing [87–89], the field that studies how generic dynamic systems can be used for universal computation.

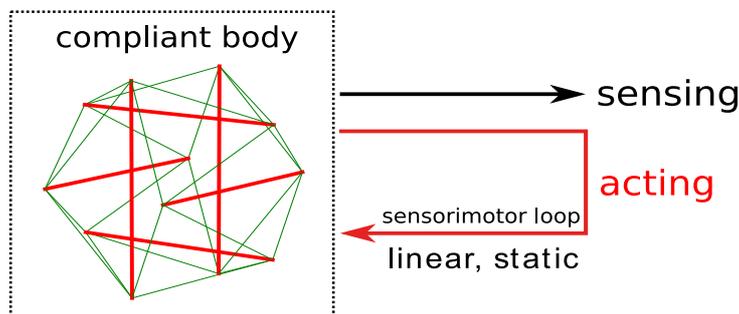


Figure 5.2: Overview of the approach.

Building on these findings, we show that Morphological Computation can be used to effectively control dynamic robot bodies. An overview of the set-up can be seen in Fig. 5.2. Note that in this work we basically only focus on pattern generation through feedback to the body. In previous work [90], we however also showed that it is possible to extract high-level environment information, such as detecting surface properties, directly and linearly from the state of the body. And this was possible even while generating locomotion patterns in parallel. This goes significantly further than the now classic demonstrations from Paul, who was the first to conceive simple robots that performed computation through the body [72].

¹With reasonable assumptions, they can be used to approximate any non-linear filter with fading memory.

Indeed, Paul also considered tensegrity structures for morphological computation, but the controller in her set up was still external [91].

The main goal of this paper is three-fold. First, we show that the results of Hauser et al. are not merely a theoretical result and show that compliant robots indeed have real computational power which can be easily exploited using simple learning algorithms. Secondly, we use as example computation, the generation of cyclic motion patterns (similar to the patterns generated by Central Pattern Generators (CPGs)), and use this to achieve locomotion. By using the morphology to generate CPG-like signals, the design of the controller can be drastically simplified. Indeed, integrating sensor data into CPGs is not an easy problem, and by integrating the body dynamics in the control structure, the robot intrinsically synchronizes to properties in its environment. Finally, by using tensegrity structures, we provide an implementation of the general principle of Physical Reservoir Computing, that is very close to the pure mass-spring nets from Hauser et al., but that is physically implementable, as they can be made free-standing.

The structure of this paper is as follows. We first provide an overview of tensegrity structures, central pattern generators and reservoir computing to make the paper self-contained. We then introduce and compare three learning rules for learning CPG-like motor patterns with tensegrity structures. Next, we provide a set of example applications. We show that the gait can be modulated by changing the equilibrium length of a subset of springs, which can prove useful to train robots to adapt their gait depending on the terrain. We optimize gaits using an external controller and then learn the equivalent gait with morphological computation to show that the control can literally be outsourced to the body. In the same spirit, we show an example of the control of an end-effector. Finally we empirically show that the presented methods work over a large parameter space and in non-linear regions when the structures are driven far from their equilibrium state.

5.2 Tensegrity structures

Tensegrities are remarkable structures consisting of compressive elements connected through tensile elements only [92, 93]². In this section we first introduce the dynamics of tensegrity structures. We then review some of the literature on tensegrities in different fields. In Appendix B we explain how we defined the spring constants and equilibrium state of the structures used in this work.

5.2.1 Tensegrity dynamics

Formally, we can define a tensegrity as a finite set of labeled points called nodes or endpoints [96]. Tensegrity structures, trusses, tensile structures and the mass-

²There is some discussion on whether Snelson [93] or Fuller [92] invented the tensegrity concept (see [94], p.221 for a discussion or [95]).

spring nets studied by Hauser et al. [85, 86] are pin-jointed structures. Here we only consider bar-spring tensegrities, in which pairs of nodes can be connected by a member which is a bar or a spring. At the end of this section we will show that mathematically mass-spring nets and tensegrity structures are similar with additional non-linearities arising in tensegrities from inertia properties and the fixed bar lengths. Springs are members resisting only tensile forces if they are stretched beyond their equilibrium length. They do not resist compression (they go slack) and for simulation purposes we neglect their mass properties. Bars are members resisting both compressive and tensile force. They do not change length.

We assume the mass of each bar to be evenly distributed along its longitudinal axis. It is further assumed that the bar is infinitely thin and as such the inertia of a bar can be described by only taking the moment of inertia around an axis perpendicular to the longitudinal axis into account. If we place a reference frame at the center of mass of the bar, then the moment of inertia is $j = \frac{ml^2}{12}$, with m the mass of the bar, and l its length.

In this work, we will only consider class 1 tensegrities [97], i.e. *pure* tensegrities. This means that no two bars ever share a common node. Furthermore, each node needs to be attached to a bar. Hence there are exactly $n = 2b$ nodes with b the number of bars.

Three-bar tensegrity prisms are the simplest class 1 tensegrities, consisting of 3 bars and 9 springs. These structures can be stacked to create snake-like structures by adding springs between the prisms. An example of such a structure is shown in Fig. 5.3. The structure is prestressed and free-standing (it does not collapse under gravity).

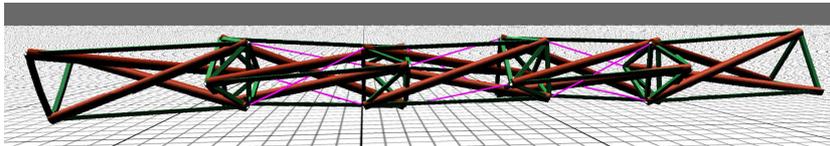


Figure 5.3: A snake tensegrity robot in our simulator, made out of 5 stacked tensegrity prisms. The red cylinders are bars (fixed length) resisting both tensile and compressive forces, the thick green lines are springs resisting tensile forces and the thin purple lines are springs with varying equilibrium lengths (actuated).

Let us now define the dynamics of a class 1 tensegrity with stiff bars and springs. The primary purpose of this section is to show where the non-linearities of the system arise from and which parameters need to be chosen when constructing tensegrity structures.

We will follow the description from [97]³. First note that because the springs

³We have simplified the notation by only considering bars with uniform mass, described by their center of mass.

only generate forces, but do not have mass, we only need to integrate the trajectories of the bars. One degree of freedom is lost for each bar, because the bar length is fixed. Hence the total number of degrees of freedom is $5b$.

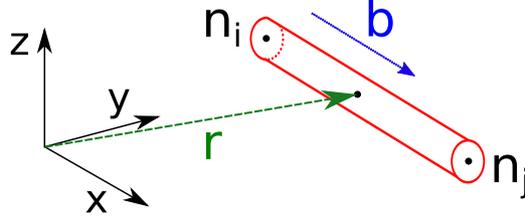


Figure 5.4: Non-minimal set of generalized coordinates for the description of a bar. The vector \mathbf{r} points to the center of mass of the bar and \mathbf{b} lies along longitudinal axis of the bar.

A description of the dynamics with a minimum number of coordinates is given in [98]⁴. Here, we use 6 generalized coordinates $\mathbf{q} = [\mathbf{r}^T \mathbf{b}^T]^T$ per bar (Fig. 5.4) as this simplifies the equations. The coordinate vector \mathbf{r} is fixed to the center of mass of the bar and \mathbf{b} is a unit vector along the longitudinal axis of the bar, the direction of which can be chosen arbitrarily.

Let the Cartesian coordinates of all nodes be given by⁵:

$$\mathbf{N} = \begin{bmatrix} x_1 & y_1 & z_1 \\ \dots & \dots & \dots \\ x_n & y_n & z_n \end{bmatrix}^T. \quad (5.1)$$

The transformation from generalized coordinates to the Cartesian coordinates is now given by:

$$\mathbf{N} = \mathbf{Q}\Psi^T, \quad (5.2)$$

where $\mathbf{Q} = [\mathbf{r}_1 \dots \mathbf{r}_b \mathbf{b}_1 \dots \mathbf{b}_b]$ and Ψ is a square, invertible matrix. If we order the nodes such that node i is connected to node $i + b$ through a bar, then Ψ has a convenient structure:

$$\Psi = \begin{bmatrix} \mathbf{I} & \mathbf{L}/2 \\ \mathbf{I} & -\mathbf{L}/2 \end{bmatrix}, \quad (5.3)$$

where the diagonal matrix \mathbf{L} contains the lengths of the bars.

As [97] shows, the generalized forces can now be obtained from the forces acting on the nodes through a linear transformation:

$$\mathbf{F}_q = (\mathbf{W} - \mathbf{N}\mathbf{C}^T \text{diag}(\boldsymbol{\lambda})\mathbf{C})\Psi \quad (5.4)$$

$$= (\mathbf{W} - \mathbf{Q}\Psi^T \mathbf{C}^T \text{diag}(\boldsymbol{\lambda})\mathbf{C})\Psi, \quad (5.5)$$

⁴ We will also consider this description in Appendix B for the feedback linearizability of a single bar attached to springs.

⁵We use the notation x to denote a scalar, \mathbf{x} for a vector and \mathbf{X} for a matrix.

where \mathbf{W} contains the external forces acting on the nodes. The matrix $\mathbf{C} \in \{0, 1, -1\}^{s \times n}$ is called the connectivity matrix and contains only ones, zeros and minus ones. Here s is the number of springs, which is at least $3n/2$ (each node is connected to at least three springs).

In this work, we only consider linear springs and therefore, the force densities λ can be written as:

$$\lambda = \max(k(1 - l_0/\|\mathbf{n}_i - \mathbf{n}_j\|), 0), \quad (5.6)$$

where k is the spring constant of the spring, l_0 the equilibrium length of the spring and \mathbf{n}_i the i th column of \mathbf{N} . Normally one should prevent the springs from going slack, as this risks collapsing the structure.

The external forces are due to ground collisions and are modeled as explained in Appendix B and damping:

$$\mathbf{W} = \mathbf{W}_{ext} + \dot{\mathbf{N}}\mathbf{R} \quad (5.7)$$

$$\mathbf{R} = \zeta \mathbf{C}^T \mathbf{C}. \quad (5.8)$$

The damping we consider only acts along the springs \mathbf{R} with a uniform damping coefficient ζ .

We then obtain the following matrix differential equation:

$$\mathbf{M}(\ddot{\mathbf{Q}} + \mathbf{\Xi}\mathbf{Q}) = \mathbf{F}_q, \quad (5.9)$$

where the mass matrix \mathbf{M} is given by $\mathbf{M} = \text{diag}([m_1 \dots m_b j_1 \dots j_b])$ (j_i are the moments of inertia of the bars) and $\mathbf{\Xi} = \text{diag}([0 \dots 0 \xi_1 \dots \xi_b])$ is a diagonal matrix describing the rotational equations of motion and the constraints $\|\mathbf{b} = 1\|$. These Lagrange multipliers are given by:

$$\xi_i = \frac{\dot{\mathbf{b}}_i^T \dot{\mathbf{b}}_i + j_i^{-1} \mathbf{b}_i^T \mathbf{F}_{q,(b+i)}}{\mathbf{b}_i^T \mathbf{b}_i}, \quad (5.10)$$

where $\mathbf{F}_{q,(b+i)}$ is the $b+i$ th column of \mathbf{F}_q .

The dynamics of a mass-spring net are obtained for $\mathbf{\Psi} = \mathbf{I}$, \mathbf{M} a diagonal matrix containing the weights of the point masses and $\mathbf{\Xi} = \mathbf{0}$. This points out that tensegrities behave similarly to spring-mass nets, but with additional nonlinearities arising from the rotational equations of motion and the bar constraints. We note that even with linear springs a 3 dimensional mass-spring net will have non-linear dynamics for non-zero equilibrium lengths. This is due to the non-linearity of the Euclidean distances in Eq. 5.6.

5.2.2 Related work on tensegrity structures

Tensegrity structures have an architectural and art background. Most of the early research on these structures focused on finding stable configurations and describing

their static properties [92, 96, 99, 100]. The result of this research is that a vast amount of typical configurations and properties thereof is now available [94, 101].

Much less literature is available on the dynamic properties of tensegrities. Motro et al. [94] list a few examples of actuated structures and Sultan et al. [102] investigated the linearized dynamics. Skelton et al. [97] provide Lyapunov function based control techniques, but the practical use of their method might be limited on underactuated robotic platforms.

Paul et al. [91] were (to the best of our knowledge) the first to link tensegrity structures to the morphological computation domain. They evolved gaits for tensegrity prisms and discusses the robustness of these robotics systems to actuator failures. Rieffel et al. [103] went a step further by introducing *morphological communication*. In their work, independent controllers for parts of a tensegrity structure interact only through the dynamics of the structure, i.e. the structure itself is used as a communication tool. More recently, Bliss [104] has shown an interesting example of taking the (linearized) dynamics into account while developing a CPG-based controller (see next Section) for a tensegrity structure.

Our choice for tensegrity structures also has a biological inspiration. Ingber has done remarkable work on cellular mechanics based on tensegrity structures [105–107]. On this level, there is no neural control and the information exchange is chemical. Because the techniques we present in this work make no assumptions on the type of actuation or sensor feedback, they might be used as a tool to explain or study the fundamental mechanisms of cell movement and mechanotransduction [108].

5.3 Central pattern generators

Central pattern generators (CPG) are neural circuits typically found in the spine of vertebrates that generate rhythmic activation patterns without sensory feedback or higher level control input [109]. Our prime goal is to show that a lot of computational power can be exploited in compliant structures. The more computations that can be outsourced to the body, the less effort one needs to put in the construction of CPGs (for robotics applications) and the less external computational power is needed.

Robotic systems are not often as compliant as the ones we study here and the available morphological computational power might be insufficient to allow for the desired behavior with a static linear feedback. We argue that one should however try to keep the body’s dynamics as much (and as soon as possible) in the loop to be able to exploit the morphological computational power. Indeed, in the compliant tensegrity structures we can go as far as leaving out the external CPG completely.

5.3.1 Matsuoka oscillators

The type of non-linear oscillator we consider in this work as a model CPG is called the Matsuoka oscillator [110, 111]. It is one of the most fundamental oscillator structures, based on a simple integrating neuron with fatigue. Its dynamics are given by (dropping the time indices):

$$\dot{\mathbf{x}}^{osc} = \frac{-\mathbf{x}^{osc} - \mathbf{A}\mathbf{y}^{osc} + \gamma - \iota\mathbf{v}^{osc}}{\tau_1} \quad (5.11)$$

$$\dot{\mathbf{v}}^{osc} = \frac{\mathbf{y}^{osc} - \mathbf{v}^{osc}}{\tau_2} \quad (5.12)$$

$$\mathbf{y}^{osc} = \max(\mathbf{x}, \mathbf{0}). \quad (5.13)$$

Here \mathbf{A} is the matrix describing how the neurons are connected. It is typically sparse (Matsuoka mostly analyzed small regular connection patterns). The positive semidefinite connection matrix \mathbf{A} was constructed similar to the stress matrix of the tensegrity structure with the diagonal (self feedback) removed. More precisely:

$$\mathbf{h} \in [0, 1]^n \quad (5.14)$$

$$\mathbf{A} = \mathbf{C}^T \text{diag}(\mathbf{h})\mathbf{C} - \text{diag}(\text{diag}(\mathbf{C}^T \text{diag}(\mathbf{h})\mathbf{C})), \quad (5.15)$$

where \mathbf{C} is the connectivity matrix as defined in Section 5.2.1. Hence, the neurons are connected in the same way as the springs connect the nodes of the tensegrity structures. The choice of this connection pattern was in a sense arbitrary. However, random connection patterns tend to generate chaotic trajectories, which are unwanted in this work.

The integrating neuron and the fatigue have time constants τ_1 and τ_2 respectively. The steady state firing rate of the neuron is determined by ι and γ is called the impulse rate of the tonic or slowly varying input [110]. In this work, we keep these parameters constant, i.e. the oscillator itself is not modulated. The parameters are: $\iota = 1, \tau_1 = 0.5, \tau_2 = 5$ and $\gamma = 1$. Fig. 5.5 shows an example of CPG signals generated by the above procedure. There were a total of 12 dimensions (5 shown) and the connection pattern was taken from the tensegrity icosahedron shown in Fig. 5.6.

\mathbf{y}^{osc} contains the firing rate of the neurons and \mathbf{v}^{osc} models the fatigue. The firing rates \mathbf{y}^{osc} are the outputs of the oscillator and these signals are used to construct the target motor signals. We resampled the output signals \mathbf{y}^{osc} such that the signals had the correct frequency for the experiment (normally 1 Hz). The desired output signals are random linear combinations of this N-dimensional signal \mathbf{y}^{osc} .

Based on the signals \mathbf{y}^{osc} we construct target motor signals as a simple linear combination:

$$\mathbf{y}^{target} = \mathbf{W}^{target}\mathbf{y}^{osc}. \quad (5.16)$$

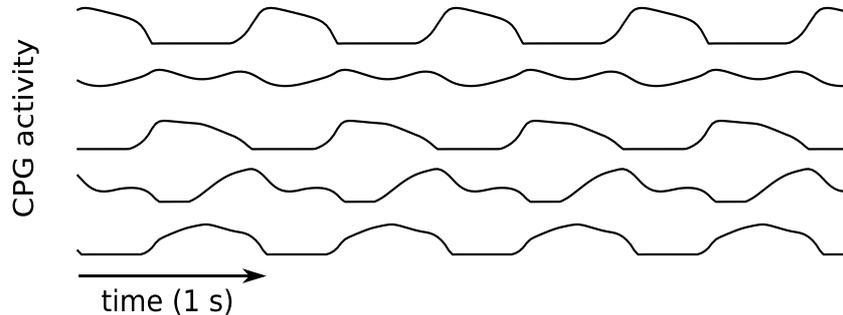


Figure 5.5: Sample Matsuoka oscillator signals. A linear combination of such signals is used as CPG signal for robot locomotion. There were a total of 12 dimensions (5 shown) in CPG of this example and the connection pattern is taken from the tensegrity icosahedron (Fig. 5.6).

In practice, we also add a constant bias variable to \mathbf{y}^{osc} . \mathbf{W}^{target} is random (normally distributed values) in most of this work (i.e. we assume the desired CPG signal to be known), except in Section 5.5.2, in which we optimize the CPG signal. Both \mathbf{W}^{target} and \mathbf{y}^{osc} are random. The fundamental difference is that \mathbf{y}^{osc} is *generated* by a random oscillator. These signals will take values between 0 and 1, while the motor signals will need a correct offset and amplitude. This is solved by \mathbf{W}^{target} , which combines the signals from \mathbf{y}^{osc} into meaningful motor commands.

We chose the Matsuoka type of oscillator because of its simple structure which can be chosen similarly to the connection pattern of the tensegrity structure itself. While we have not yet explored this path, we hope the morphological communication idea from Rieffel et al. [103] can be integrated in this way.

5.4 Physical Reservoir Computing

Classic techniques for training recurrent neural networks, such as backpropagation through time [112], approximate a desired output signal by modifying the internal weights of the neural network (as well as the readout weights, if any). This is often cumbersome and difficult to implement correctly as one needs gradient information to apply the chain rule. Furthermore, backpropagation through time is prone to local minima.

Reservoir Computing (RC) is a conceptually much simpler technique to train such recurrent neural networks [89]. Instead of modifying the internal weights, the original network is left as-is and only the readout weights are modified. The original network essentially becomes a *computational black-box*. The outcome of this training procedure typically depends on a few parameters that define the regime of the neural network.

Reservoir Computing (RC) is known under different terms, depending on the

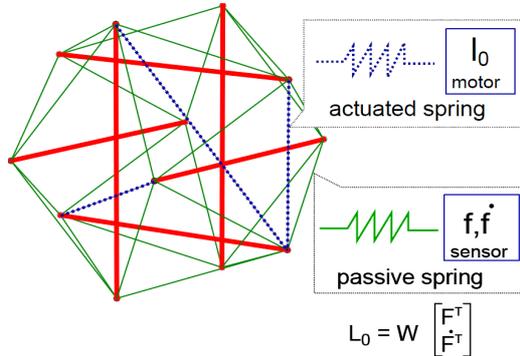


Figure 5.6: Overview of physical reservoir computing with compliant tensegrity structures. The thin green lines are passive springs connected to a sensor measuring the force and its derivative on the spring. The thick red lines are bars, non-compliant compressive members. The dotted lines are actuated springs, implemented as a passive spring connected to a motor modifying the equilibrium length of the spring. The (new) equilibrium length of the actuated springs is computed as a linear combination of the sensor values. There are no joints in the system and the only constraints are the fixed bar lengths.

type of recurrent network that is trained. Most importantly we distinguish Liquid State Machines [88] and Echo State Networks [87, 113]. The core idea of RC, originally applied only to neural networks, was since extended to other non-linear dynamical systems, leading to what we call Physical Reservoir Computing (PRC). There have been demonstrations of the reservoir computing approach applied to different domains such as photonics [114] and more abstractly electronics [115]. All these implementations share the common idea that a system with complex dynamics is perturbed externally but left untouched otherwise, and a simple readout mechanism is trained to perform the desired computational task. While the idea of PRC originated in the context of neural networks, recent theoretical results have extended the applicability of this computational framework immensely, showing that any dynamical system of a given size, obeying easily satisfied constraints, has the same computational power [116].

Let us start with the most common and straightforward implementation of Reservoir Computing, namely Echo State Networks in combination with a linear readout layer⁶. The discrete time network dynamics are given by:

$$\mathbf{x}[d+1] = \tanh(\mathbf{W}_{res}\mathbf{x}[d] + \mathbf{W}_{in}\mathbf{u}[d+1]) \quad (5.17)$$

$$\mathbf{y}[d+1] = \mathbf{W}_{out}\mathbf{x}[d+1]. \quad (5.18)$$

⁶Often a bias input is added which makes the equations non-symmetric. The non-linearity \tanh is most often used, but variants are possible.

There are two main applications of such a system. First one can use it to approximate non-linear filters by training the readout \mathbf{W}_{out} . In this case one normally scales \mathbf{W}_{res} such that the system has the fading memory property (based on the spectral radius). Simply stated, this means that when the input is removed, the system dynamics will die out.

Secondly, RC can also be used to implement function that do not necessarily have the fading memory property [117]. This can be achieved by feeding the output back into the system:

$$\mathbf{x}[d+1] = \tanh(\mathbf{W}_{res}\mathbf{x}[d] + \mathbf{W}_{in}\mathbf{u}[d+1] + \mathbf{W}_{fb}\mathbf{y}[d]) \quad (5.19)$$

$$\mathbf{y}[d+1] = \mathbf{W}_{out}\mathbf{x}[d+1]. \quad (5.20)$$

The feedback weights \mathbf{W}_{fb} are typically chosen randomly and again, only \mathbf{W}_{out} is trained. This system can be used to autonomously generate desired signals.

The first kind of task is clearly easier to train, as it is an open loop system. For signal generation tasks, small changes to the feedback weights can have a large influence. To imitate CPG signals with morphological computation, we need to consider the second approach.

Fig. 5.6 shows how we used tensegrity structures for physical reservoir computing. The force and its derivative on each spring are sampled and used as input \mathbf{x} . The equilibrium length of a subset of the springs is used as feedback to the system. Differently from [85], we use only linear springs (Eq. 5.6). In our experiments, the non-linearities arising from the changing geometrical configuration and inertia are sufficient for good performance. We now define the system state (cf. Eq. 5.19) for our setup:

$$\mathbf{x}(t) = \text{vec} \begin{pmatrix} \mathbf{f}(t) & \dot{\mathbf{f}}(t) \\ \mathbf{f}(t-\Delta) & \dot{\mathbf{f}}(t-\Delta) \\ \dots & \dots \\ \mathbf{f}(t-k\Delta) & \dot{\mathbf{f}}(t-k\Delta) \end{pmatrix}, \quad (5.21)$$

where $\mathbf{f}(t)$ are the spring forces measured at time t . $\Delta = 20ms$ is the controller time step and k is the number of delay steps used. For the tensegrity icosahedron simulations, we used $k = 9$ (maximum delay of 200 ms) and $k = 3$ for the snake robots. The main rationale for this is that this allows the feedback to filter out noise due to ground collisions to some degree (by averaging over the delayed inputs)⁷. One can see from Fig. B.1 that the time delayed sensor information is indeed highly correlated. Using Hooke's law and Eq. 5.6, each element of $\mathbf{f}(t)$ can be written as:

$$f_e(t) = \max(k_e(\|\mathbf{n}_i(t) - \mathbf{n}_j(t)\| - l_{0,e}(t)), 0). \quad (5.22)$$

⁷We also did a number of simulations with $k = 0$ (see also [90]) to verify that the system does not depend fundamentally on this delayed sensor information. This was indeed not the case, but the ground collisions tend to render figures such as Fig. 5.8 less intelligible.

We explicitly used the time index for the equilibrium lengths $l_{0,e}(t)$, because the tensegrity structures we consider contain springs with varying equilibrium lengths. We shall call the subset of the springs of which the equilibrium length $l_{0,e}$ can be modified actuators, actuated springs or motors. Now, we call the subset of passive, fixed equilibrium length springs *pas* and the subset of actuated springs *act*. \mathbf{l}_0^{pas} is a constant vector defined by the equilibrium state of the structure. $\mathbf{l}_0^{act}(t)$ is time-varying and is given by:

$$\mathbf{l}_0^{act}(t) = l_{max}g(\mathbf{y}(t)) + \mathbf{l}_0^{act}(0). \quad (5.23)$$

l_{max} is the maximum change in equilibrium length of the springs (w.r.t. the initial lengths in $\mathbf{l}_0^{act}(0)$) allowed by the actuators. For this to work we must have $g : \mathbb{R}^a \rightarrow [-1, 1]^a$, with a the number of actuated springs. Now $\mathbf{y}(t)$ will in general be a linear combination of $\mathbf{x}(t)$ and a constant bias input:

$$\mathbf{y}(t) = \mathbf{W}\mathbf{x}(t). \quad (5.24)$$

The goal of most of the algorithms we will study is to optimize the matrix \mathbf{W} .

In the experiments presented in this article, we used $g(\mathbf{y}(t)) = \tanh(x)$. It is important to justify the use of a non-linear function, as it can provide computational power (as in the RC approach). Therefore, we also tested the setup with both a hard limit: $g(\mathbf{y}(t)) = \min(\max(\mathbf{y}, -1), 1)$ and with the identity function (no limit). Both cases, provided quantitatively similar results to those presented in the experimental section 5.5. The identity function was discarded because it does not guarantee boundedness of the feedback and spurious sensor data can make the structures collapse. In practice we noticed that with the identity function, the structure would operate correctly for e.g. 30 s after training and then collapse because of an extreme sensor value during a ground collision. As explained in B, ideal motors were assumed. However, a physical implementation will always be limited by the maximum offset of the motor, which validates the use of $g(\mathbf{y}(t))$.

To conclude this section, we note that for our setup $\mathbf{x}[k]$ from Eq. 5.19 is replaced by sensor measurements from the tensegrity structure and the output $\mathbf{y}(t)$ is a linear combination of these values. Differently from the classic RC or ESN implementation, the feedback enters the system through a physical modification of the system by modifying the equilibrium lengths of a set of actuated springs. The system itself is continuous time, but the spring lengths are only updated at discrete time steps.

5.5 Experiments

The experimental section of this paper consists of three parts. First, we introduce a set of algorithms to train tensegrity structures to produce rhythmic patterns. Next, we discuss possible applications for locomotion. We end with a comparison of different parameter combinations to study the importance of non-linearities in the system.

5.5.1 Outsourcing motor pattern generation

Recursive Least-Squares approach

The first training algorithm we will consider is based on the Recursive Least-Squares (RLS) algorithm [118]. When the same samples are presented to the RLS algorithm, it will compute the same weights as batch linear regression (which we used in previous work [90] and is also used by [85]). The advantage of RLS are that it allows us to gradually transition from a completely teacher forced structure (the desired signals are fed into the system) to a system generating its own control signals and to restart training if needed.

There are two disadvantages in our opinion. First, one needs to update the matrix containing the covariances of all the input variables, which does not scale well.

The second and more fundamental disadvantage is the dependence on explicit knowledge of the target function, because one needs to know the difference (error) between the optimal motor signal and the current signal generated by the RLS algorithm. In a practical setting we do not always know the target signal and often only have some global performance measure at hand.

We now describe the RLS algorithm in detail. During training the output signal is a mixture of the target output signal and the feedback output signal which is being trained. The influence of the target signal on the output signal is gradually reduced until the output signal is given by the trained feedback only:

$$\alpha_{rls} = \frac{1}{1 + \tau_{rls}t} \text{ if } t < \text{train time else } 0 \quad (5.25)$$

$$y_i(t) = \alpha_{rls} y_i^{target}(t) + (1 - \alpha_{rls}) \sum_j W_{i,j}^{rls}(t) x_j(t). \quad (5.26)$$

At each time step the weights \mathbf{W}^{rls} are updated using the RLS equations:

$$\mathbf{L}^{rls}(t) = \frac{\mathbf{P}^{rls}(t)\mathbf{x}(t)}{1 + \mathbf{x}(t)\mathbf{P}^{rls}(t)\mathbf{x}(t)} \quad (5.27)$$

$$\mathbf{P}^{rls}(t + \Delta t) = \mathbf{P}^{rls}(t) - \frac{\mathbf{P}^{rls}(t)\mathbf{x}(t)\mathbf{x}^T(t)\mathbf{P}^{rls}(t)}{1 + \mathbf{x}^T(t)\mathbf{P}^{rls}(t)\mathbf{x}(t)} \quad (5.28)$$

$$\mathbf{e}^{rls}(t) = \mathbf{y}^{target}(t) - \mathbf{W}^{rls}(t - \Delta t)\mathbf{x}(t) \quad (5.29)$$

$$\mathbf{W}^{rls}(t) = \mathbf{W}^{rls}(t - \Delta t) + \mathbf{L}^{rls}(t)\mathbf{e}^{rls}(t). \quad (5.30)$$

There is only a single parameter, namely the teacher forcing decay time constant τ_{rls} . The covariance matrix \mathbf{P}^{rls} was initialized using the identity matrix. We note the difference from FORCE learning [119] in which initially chaotic systems are used. The main reason for this is that tensegrity structures are inherently damped and to create chaos, one would need a feedback loop to drive the system. From a practical point of view this might be inefficient, as one would need additional actuators which are only used to keep the system active. In this sense, the

RLS approach used here is closer to the teacher forcing approach [120]. In this approach the desired output is fed into the system during training and the state of the system $\mathbf{x}(t)$ is stored. Then, regression is used to approximate the desired output from the system state. Finally, during testing the approximate output based on the system state is fed back into the system and the system will generate the desired patterns autonomously. The testing phase is also called free run, as the system is not longer forced by the external input.

The gradual change from teacher forcing to free run used in this work allows the structure to take over the control in a smooth way and to restart learning in a straightforward way. We noticed that the RLS algorithm becomes unstable if learning continues with α_{rls} too low (i.e. $\pm\alpha_{rls} < 0.03$). So we simply switch to free run when α_{rls} drops below the threshold. The most likely explanation for the instability is that this is caused by the phase drift between the output and the teacher signal when the system is unforced.

A demonstration of the RLS approach is shown in Fig. 5.7 and Fig. 5.8. In this case 6 actuators were used (i.e. 6 output dimensions). One can observe that the output signal gets out of phase with respect to the target signal, due to collisions with the ground. There is noise in the system, due to the control time step (20 ms) and the ground collision.

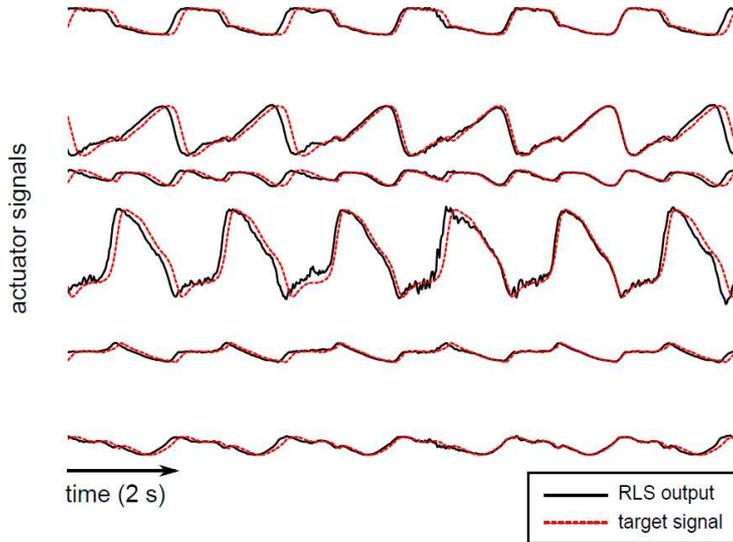


Figure 5.7: Demonstration of the RLS algorithm. 6 outputs were trained for 250 s, followed by 150 s of testing. Shown is the output at the end of the testing phase. The dotted line is the target signal, which is generated as in Eq. 5.20. The full line is the output signal, which is sent to the actuators. Notice that the phase of the target signal is not matched, but that the relative phase of the outputs is fixed. This effect is due to the tensegrity structure synchronizing to its collisions with the ground.

Fig. 5.8 shows a phase portrait of two output signals from Fig. 5.7. The output signals stay in phase w.r.t. each other, which is important for locomotion. The RLS rule can capture the complex details of the target signals through the non-linearities provided by the structure.

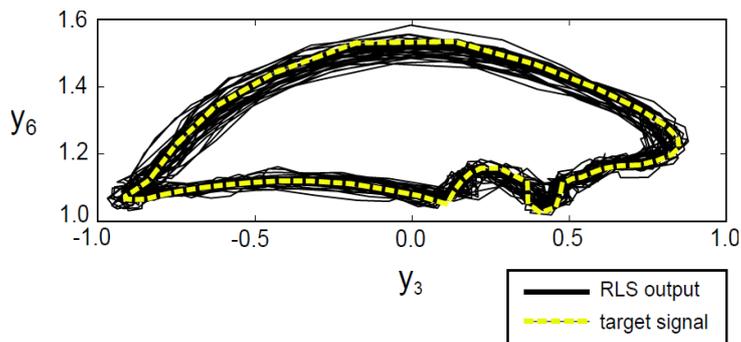


Figure 5.8: Demonstration of the RLS algorithm as in Fig. 5.7. Shown are two output dimensions out of 6 in total during 20 s of testing. The light line is the target signal, the dark line the output signal. Clearly the system has learnt the attractor robustly. The small perturbations are mostly due to ground collisions.

One might ask if the system is not simply vibrating along one of its shape modes? Such a result would not be useful as for locomotion tasks we want the system to undergo large shape deformations. The shape modes of the tensegrity icosahedron which was used in this experiment (without the actuators) can be found in [100]. We show that this is not the case in Fig. 5.9. In this example we simulated a tensegrity in free fall to prevent collisions and again trained a random motor pattern with 9 actuators. The complex trajectories of the endpoints of each bar are shown.

Gradient descent approach

To overcome the first disadvantage of the RLS algorithm, namely its complexity vis-à-vis its biological plausibility, we use stochastic gradient descent on the error signal. The following equation is obtained easily by differentiating the quadratic error at a time step. We can then replace the update of \mathbf{W}^{rls} with:

$$\mathbf{W}^{gd}(t) = \mathbf{W}^{gd}(t - \Delta t) - \alpha^{gd} \mathbf{e}^{rls}(t) \mathbf{x}^T(t). \quad (5.31)$$

Because the learning rate α^{gd} has to be chosen small enough to prevent instability, the GD rule converges slower in practice than the RLS rule.

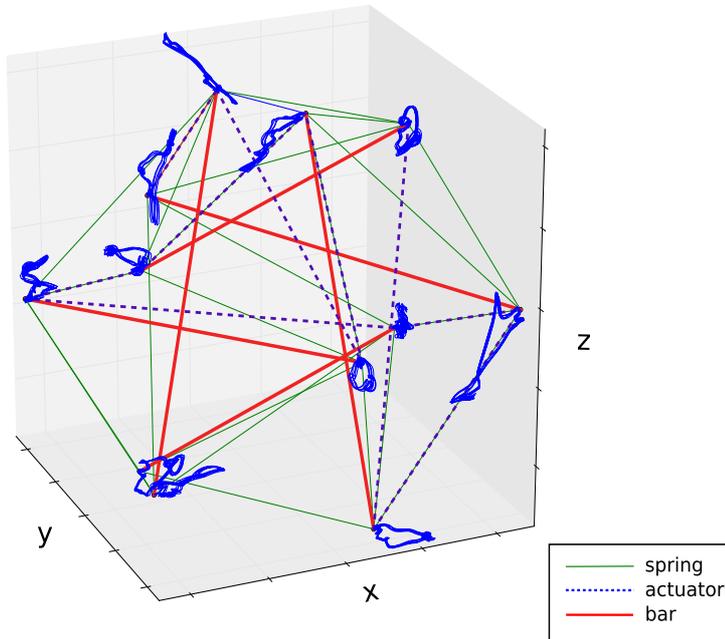


Figure 5.9: Complex motor motion patterns learnt by the tensegrity structure, based on random CPG signals. Shown in blue are the trajectories of the endpoints of each bar. The bars are red, springs are green and actuated springs are dashed lines (9 actuators). Compare with the shape modes from e.g. [100].

Eliminating the teacher: reward modulated Hebbian approach

For various reasons one might prefer to use only a single reward signal instead of having an error function per output. We might not be able to conceive a suitable error function, when for example an error measure is only available at a non-actuated spring. It is also more biologically plausible to have only a limited amount of reward signals. We will only consider instantaneous reward, but extensions are possible by using eligibility traces [121].

Note that the essence of the gradient rule is that the weight changes should follow the correlation between the input variables and the error signal. A reward can be interpreted as the inverse of an (absolute) error. The reward is large (in amplitude) when the error is small and vice versa. So instead of doing gradient descent on the error signal, we can equivalently do gradient ascent on the reward signal.

The meaning of a *large* reward is ambiguous, because to replace $e^{rls}(t)$ with

some measurement of the reward, we need the scalars in this vector to take positive values when reinforcing the weights to this output would increase the reward and vice versa, while a reward can have an arbitrary offset ⁸. So the trick is that we need to subtract the baseline performance from the reward, or stated differently we need to know how surprising a reward is [123].

We often do not know the reward signal explicitly. Hence, we cannot find an analytic form of the derivative of the reward. The trick to overcome this is to use finite differences to estimate the derivative of the reward. For this we add random noise to the output and observe changes in the reward signal. The reward signal is usually one-dimensional, so we need to find out which weight should be reinforced.

Legenstein et al. recently used a learning rule based on these observations for training relatively large (compared to our tensegrity structures) neural networks [124]. In addition, they also assumed the noise signal to be unknown and estimated the noise from the system output. We assume the noise signal to be known and use a reward modulated Hebbian learning rule similar to the one from [121, 125, 126]. [125] also provides non-Hebbian variations on this rule.

The reward modulated Hebbian rule (RMH) we used is given by:

$$\mathbf{W}^{rmh}(t) = \mathbf{W}^{rmh}(t - \Delta t) - \alpha^{rmh} \boldsymbol{\nu}(t)(R(t) - \bar{R}(t))\mathbf{x}^T(t), \quad (5.32)$$

where $R(t)$ is the (instantaneous) reward and $\bar{R}(t)$ the short term average (baseline) of the reward. The short term average reward $\bar{R}(t)$ was computed by taking the average of the rewards during the last 100 ms. The reward $R(t)$ should be a monotonic function of the error, i.e. the reward should decrease if the error increases. For the injected noise $\boldsymbol{\nu}(t)$ Gaussian white noise (GWN) was used with a decreasing standard deviation as function of time. The noise $\boldsymbol{\nu}(t)$ is not only used to update the weights, but it is also fed into the structure ($\mathbf{y}(t) = \mathbf{W}^{rmh}(t - \Delta)\mathbf{x}(t) + \boldsymbol{\nu}(t)$). Indeed, if this were not the case, one would need some critic that provides rewards based on hypothetical motor outputs. This rule reinforces weights based on how the reward and the inputs co-vary, which is why this rule is called Hebbian-like. It is similar to the classic Hebb rule but for reward and neural activity [127].

Legenstein's et al. EH-rule is given by:

$$\mathbf{W}^{rmh}(t) = \mathbf{W}^{rmh}(t - \Delta t) - \alpha^{rmh}(\mathbf{y}(t) - \bar{\mathbf{y}}(t))(R(t) - \bar{R}(t))\mathbf{x}^T(t), \quad (5.33)$$

where $(\mathbf{y}(t) - \bar{\mathbf{y}}(t))$ approximates $\boldsymbol{\nu}(t)$ under the assumption that $\mathbf{y}(t)$ varies smoothly.

The learning rule from Eq. 5.32 or Eq. 5.33 can be used in two ways. First we can simply use it to replace the RLS or GD rules, when outsourcing the computation to the structure. In this case we still need some teacher to drive the system during learning, which limits its practical use and it is more or less a replacement

⁸In fact this is not entirely true, because one can restrict the learning rules to reinforcements only as in [122], but we allow for both positive and negative weights here.

for the GD rule. Secondly, we can use it to train feedbacks without knowledge of the target signal at the neural level.

To apply RMH or similar techniques in a recurrent neural network, one typically starts from a chaotic network [119, 122] and the trained feedback drives the network toward a cyclic attractor. However, it is reasonable to assume that for robotics applications chaotic movements might be undesired. Therefore, we took a slightly different training approach. We first trained the system (using RLS) to maintain an oscillatory pattern while noise was injected through additional actuators. Hence, we obtained robust but not chaotic patterns. There are however variations in the oscillations caused by the injected noise. Then, learning through RMH starts on the additional actuators.

One might argue that the use of RLS at this point negates the advantage of RMH. However, RLS is only used to keep the system active during RMH learning and the target signals of RLS and RMH are independent (except for the fundamental frequency). A simple oscillator (e.g. a sine wave or coupled neurons) could also be used instead of a trained feedback controller. In a typical RC setup (with hyperbolic tangent neurons), it is possible to scale up the connection weights to start the learning process in a chaotic regime. In the case of tensegrity structures, we tried using a random feedback loop which we then scaled to find a chaotic regime. Unfortunately, while doing this the structures often collapsed or did not stay active and we thus concluded that this method would be cumbersome on a real platform.

The presented approach can be useful in robotic applications in which there is already some oscillatory behavior in the system. This can for example be generated by a very simple CPG signal. The RMH algorithm can then directly be applied to e.g. refine the motion. Hence, it is one possible application of the combination of a simplified CPG with our approach. The basic movements can also be provided by a controller based on linearized dynamics, where again RMH can be used to optimize the match between the actual plant and the linearized model.

In Fig. 5.10 three major phases of training using the reward based technique are shown. Here 2 feedbacks were trained using the RMH rule (on a total of 8). RLS was used to train a random motion pattern (with the same frequency) on the 6 first outputs during 200 s (left figure). Then RMH learning starts and initially the target signals are not at all matched. During training (center figure) the outputs start to match the desired signal more closely, yet there is still some visible error. During testing (right figure), the noise source is disabled and the output almost exactly matches the desired signal. In this example, the tensegrity was in free fall to remove the disturbances from ground collisions to show that the desired signal can be closely matched.

Fig. 5.11 shows a phase portrait of the two trained outputs during 40 s of testing compared to the desired output. The target signal is almost perfectly matched. In Fig. 5.12 it is shown how the RMH rule is performing gradient ascent on the reward signal. The signals were smoothed over 2 s to show the evolution of

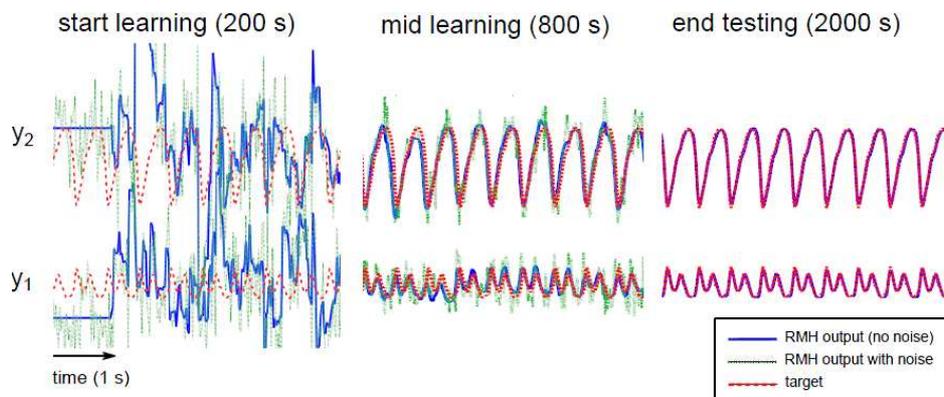


Figure 5.10: The RMH algorithm during training and testing. Training of the two RMH feedbacks starts after 200 s of training with RLS to maintain activity in the system. The tensegrity was in free fall to clearly show the difference between the three phases without influence from ground collisions. A random 6-bar tensegrity was used. The exploration noise decreased linearly as a function of time.

the reward. The figure on the right shows the reward signal with the (estimated) baseline removed. For convergence, the (short-term) mean should approximate zero as otherwise the magnitude of the weights will continue to rise or oscillate. For Fig. 5.11 an informative reward was used, namely the sum of absolute errors of both signals:

$$R(t) = - \sum_i |y_i(t) - y_i^{target}(t)|. \quad (5.34)$$

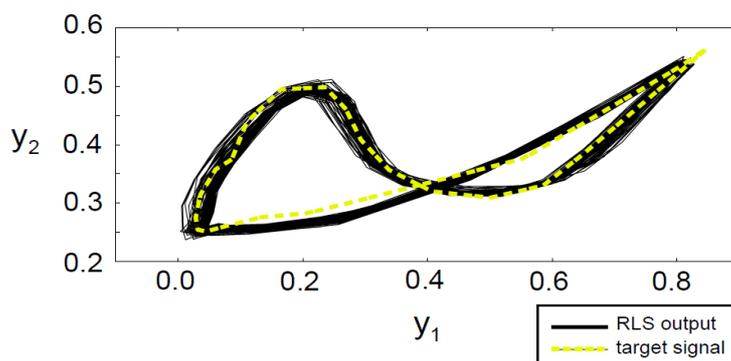


Figure 5.11: Plot of the 2 trained outputs with the RMH algorithm. The system was trained for 2000 s. Yellow: the target signal. Black: the output signal during the last 40 s of testing.

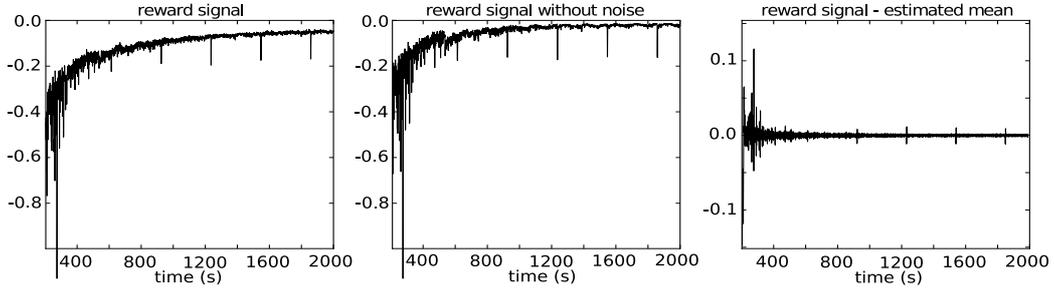


Figure 5.12: The reward modulated Hebbian algorithm performing gradient ascent on the reward. The signals were smoothed by averaging over 2 s. From left to right: reward signal, reward signal minus its short term average, reward signal without exploration noise. The reward signal minus the short term average should approach 0 to assure convergence of the weights.

Such an informative reward signal as in Eq. 5.34 need not be available for the RMH or EH rule to work. In Fig. 5.13 we applied a delta rule version of the EH rule (the noise is estimated) with a less informative reward signal:

$$\mathbf{W}^{rmh}(t) = \mathbf{W}^{rmh}(t - \Delta t) - \alpha^{rmh}(\mathbf{y}(t) - \bar{\mathbf{y}}(t))\text{sign}(R(t) - \bar{R}(t))\mathbf{x}^T(t) \quad (5.35)$$

$$R(t) = -\max_i |y_i(t) - y_i^{target}(t)|^2. \quad (5.36)$$

The result is clearly less precise than with the RMH rule (Fig. 5.11) and learning is slower. However, no knowledge of the noise is used in the learning rule and the information contained in the reward signal is limited. The delta version can further reduce the computational/communication power needed as only a single binary signal needs to be exchanged. In Appendix B we provide a method to reduce the communication load from the sensors to the motors.

5.5.2 Applications

In this section, we present a set of practical applications of morphological computation in tensegrity structures. We first show that the structure can modulate its gait patterns when we change the equilibrium length of a few springs. Next we look at gait optimization. We optimize the gait pattern with an external controller and then outsource the resulting gait to our static, linear controller. Finally, we discuss a basic end-effector control application.

Modulating motor patterns

An important question is if the trained tensegrity structures can react by adapting their gait to different configurations of the structure or e.g. the slope of a hill? To test this we added a single input signal in the system. This signal was fed into the

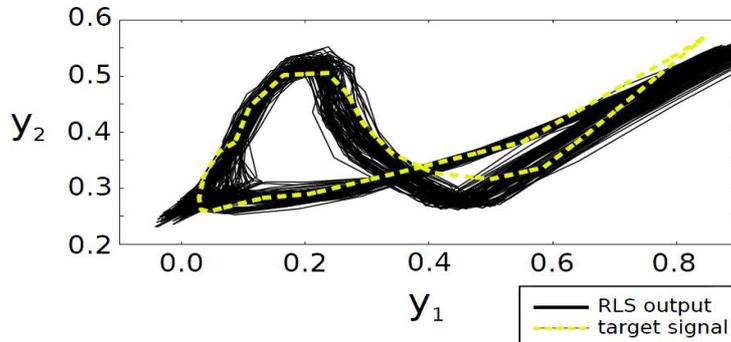


Figure 5.13: Plot of the 2 trained outputs with the EH rule and a less informative reward signal (Eq. 5.36). Hence no knowledge of the target signal, the noise or the precise reward is used. The system was trained for 3000 s. Yellow: the target signal. Black: the output signal during the last 40 s of testing.

tensegrity structure by modifying the equilibrium length of 2 actuated springs. The target motor patterns had to be modulated by the structure to linearly interpolate between two CPG patterns with the same frequency.

We again used the tensegrity icosahedron to show that such modulation is possible even in relatively small systems. Fig. 5.14 shows a result from a run of the algorithm. We trained the system for only 400 seconds. At each time step, the system switches to another random input with probability 0.005. So the time between gait changes is variable. This also shows the robustness of the system, because accidental fast switches between input states disturb the system.

Gait optimization

Gait optimization in robots is a complex problem, because small changes to e.g. the relative phase of two limbs or the duration of support phases can result in different locomotion patterns or failure in legged robots (see e.g. [1] for reviews of animal gait patterns). One for example typically positions its legs during locomotion to reduce the magnitude of joint moments and as such the required muscle forces [128].

Optimizing all aspects of gait properties is beyond the scope of this article. We assume the robot's configuration to be known, as well as the CPG frequency. Fig. 5.15 gives an overview of the training procedure we will follow. Our goal will be to optimize the weights of the matrix \mathbf{W}_{target} for a given basic CPG. We will then outsource the optimal gait to the structure with the RLS algorithm. As we will see, the obtained gaits using only morphological computation match those from during training. So the structure can approximate the required motor patterns well enough for locomotion.

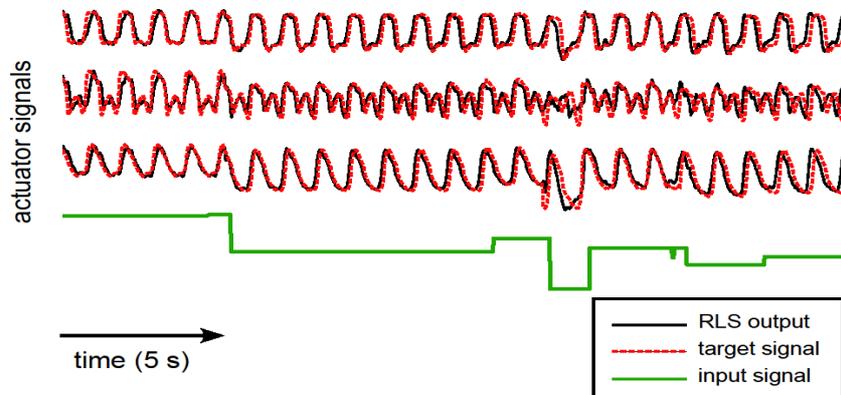


Figure 5.14: Modulating gait patterns through morphological computation. A single input signal was applied to the system by modifying the equilibrium length of two springs. The structure had to linearly interpolate between two CPG signals (3 dimensional) with the same fundamental frequency. The system was trained for 400 s with random inputs. The phase is not perfectly matched, because fast input changes disturb the system. Note that both the signal offset and shape are changed. A random tensegrity with 6 bars was used.

To optimize \mathbf{W}_{target} we use the well known CMA-ES algorithm [129]. The reason for this is that it is almost parameter free and has very good performance. The fitness function we use is simply the distance travelled by the center of mass of the tensegrity. Because of the compliance of the tensegrity structures, we do not need to include penalties for e.g. falling.

Fig. 5.16 shows the trajectory of the center of mass of three different tensegrity structures. On the left, the tensegrity icosahedron with a number of additional springs. Remarkably, the gait was obtained after only 10 iterations of the CMA-ES algorithm. The population size was 50 and there were 4 actuators. The gait was evaluated during 30 seconds. This means that only 4 hours of exploration time would be necessary to obtain this locomotion pattern on a real robot.

The two other plots are from snake-like tensegrity structures which were constructed by stacking tensegrity prisms. Fig. 5.17 shows the center structure in action, while Fig. 5.3 is the tensegrity from the right in our simulator.

To show that the same gait is indeed maintained, we compared (Fig. 5.18) the (vertical) ground reaction forces on the endpoints during training and testing of the large snake-like tensegrity from Fig. 5.17. This system has 20 actuators in total. Due to the complexity of the structure, there is some variation in the ground reaction forces, but there is a clear pattern. The relative phase between of the ground contacts is identical during training and testing. The training sample is taken from the beginning of the training (almost completely teacher forced), while the testing sample is from the end of testing (free run).

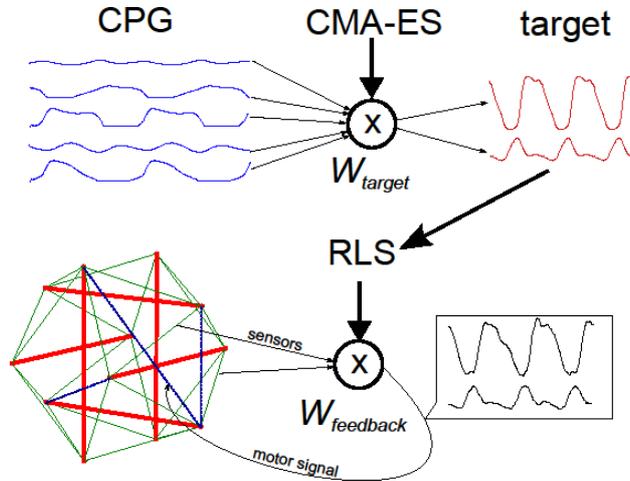


Figure 5.15: Overview of the training principle for gait optimization. We use CMA-ES to optimize the CPG pattern and then apply RLS to train a feedback to approximate this target pattern using morphological computation. If the robot has rich enough dynamics, the same gait will be obtained after outsourcing all the computations to the body.

End-effector control

To end this applications section, we show that the same technique can also be used to control an end-effector. The objective is now to control the position of the endpoint of a bar with respect to two other endpoints. For this we measure the length along two springs connecting the endpoint of the bar with the endpoints of the other bar. Imagine controlling the position of the wrist with respect to the shoulder.

We assume no model of the system is known and use CMA-ES to optimize W_{target} . The CPG has the same frequency as the target movement. Because the CPG has only a limited amount of basis signals and the structure is underactuated, it is to be assumed that the target trajectory cannot be perfectly matched. In this example we used a 30 dimensional CPG, based on a connection pattern from a stacked tensegrity prism.

To compute the fitness, we simulated the system for 100 s and computed the MSE over the last 80 seconds. The system was in free fall and the springs along which we measured the position were not actuated. A tensegrity icosahedron with a total of 13 actuators (Fig. 5.19) was used for this example (24 DOF, because the rigid body movements are ignored).

The result is shown in Fig. 5.20. While the target trajectory cannot be perfectly matched due to underactuation and the limitations of the CPG, the result is very encouraging. The end-effector is part of the computational system itself and the springs along which the position is measured influence the system as well.

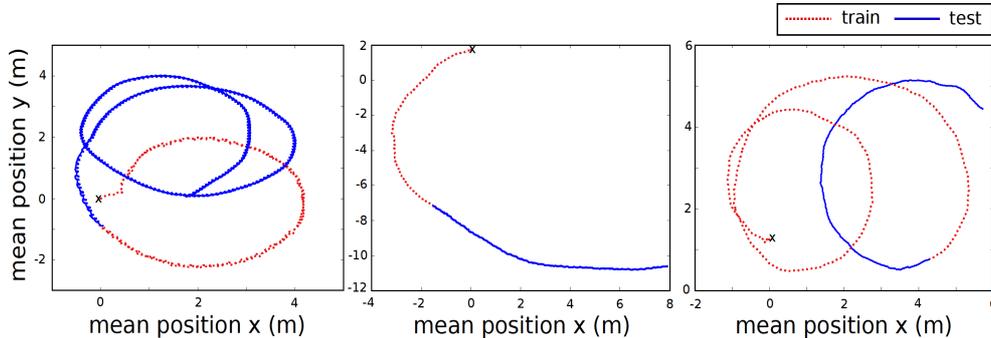


Figure 5.16: Robot trajectories (center of mass) for three runs of the algorithm on different structures (see text). In red (dotted) the trajectory during training, in blue the trajectory during testing. Morphological computation is powerful enough to maintain the same gait which was found by optimizing the external CPG.

We only used 75 s of training using RLS to transfer the control from the external CPG to morphological computation.

5.5.3 The importance of complex dynamics

To complete this experimental section, we want to show that the non-linearities can indeed improve the computational power of the system. Such a statement is of course task-dependent, e.g. to generate sine waves, it is obviously not advantageous to have complex non-linear dynamics in the system. We will again consider the generation of CPG-like signals based on the Matsuoka-type non-linear oscillator in combination with the tensegrity icosahedron.

As Sultan et al. [102] indicate the linearized dynamics of tensegrity deviate more from the non-linear dynamics of the system at higher (generalized) velocities and lower pretension (i.e. when the system is more flexible or compliant). While typically, one would restrict the velocities and deformations of the system such that the linearized dynamics are a good model of the system, our technique benefits from the opposite.

Many parameters of the structure can be tuned and optimizing the configuration of the structure itself is a daunting task. In this section we only consider the importance of two parameters, the oscillator frequency and the maximal change of the actuator equilibrium length. One can easily define physically plausible regions of operations for both parameters (see Appendix B) and we would like to know if within these regions of operation, there are significant changes of computational performance.

The task we consider is again the simulation of 12 dimensional random Matsuoka-type non-linear oscillators. The tensegrity icosahedron with a random of actuator

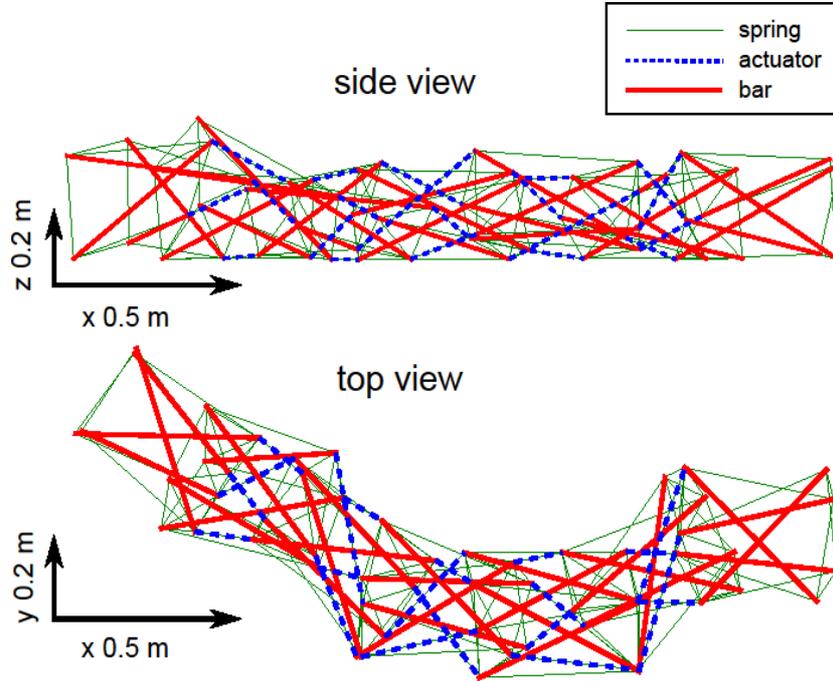


Figure 5.17: A complex snake-like structure controlled using morphological computation. Shown is the structure during locomotion with one of the found gait patterns. There are large shape deformations from the equilibrium state and a total of 20 actuators.

is used, varying from 4 to 8 motors. We swept the frequency in steps of 0.1 Hz from 0.1 to 3 Hz. The maximum spring equilibrium offset (l_{max}) was varied in steps of 3.5 cm from 5.5 cm to 37 cm. For each tuple (frequency, distance), we performed 50 trials, for a total of 15000 trials. We computed the normalized mean-square error, defined as:

$$NMSE = \frac{(\mathbf{x} - \mathbf{y})^T(\mathbf{x} - \mathbf{y})}{N\sigma(\mathbf{y})}, \quad (5.37)$$

with N the number of samples, \mathbf{x} the vectorized output and \mathbf{y} the vectorized target signal. For each set of 50 trials, only the 30 best are kept to prevent failures (e.g. collapsing) from influencing the results. The results are shown in Fig. 5.21.

So what can we learn from this? First, we see that for the task at hand, it is advantageous to work in a non-linear region by increasing the frequency of the oscillator or the maximum spring equilibrium offset. It is important to note that although the frequency is a determining factor, the technique is not constrained to the natural frequency of the system. There is broad region of frequencies with similar performance. One might consider the bottom right region of operation, with only very small amplitudes. The practical use of this region is however

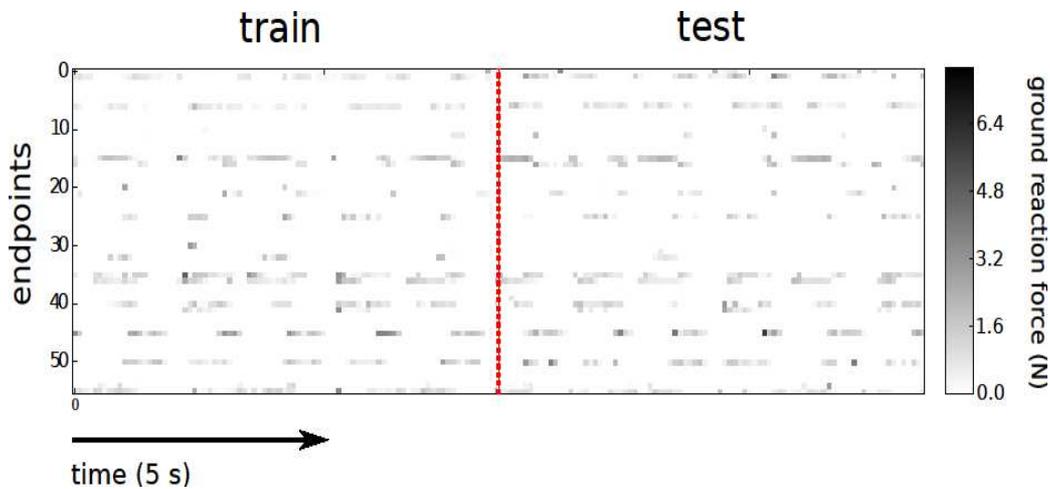


Figure 5.18: Vertical ground reaction forces during training (left) and testing (right) for the structure from Fig. 5.17. Note the variation in the signals. The same gait is maintained during testing.

limited, as the movement of the robot will be very limited.

On the other hand, going beyond the 30 cm range, often causes instability (collapsing) and in practice will cause bars to collide. In practice, the performance, will be restricted by a diagonal line going down from near the top left to the bottom right, because of practical limitations such as motor output power. So within this region, better performance can be obtained by increasing the frequency or the maximum spring equilibrium offset.

Interestingly, for the lower frequency range (which might be interesting for energy efficiency reasons) it is advantageous to increase the maximum offset. Larger deformations of the structure cause the error to decrease.

5.6 Discussion

Compliant robots have been of interest to the robotics community for over a decade. We have seen many exciting examples of very simple control laws leading to complex behavior and of robustness against external perturbation. Compliance offers multiple advantages over classic, stiff robotics: it can allow for safer robot-human interactions, increased energy efficiency, robustness against external perturbations and simplification of the control.

Notable examples of compliant robots that have very simple control laws are Puppy [16], Reservoir Dog [130], Wanda [131] and recently [132]. The control of the Reservoir Dog in irregular and unknown terrain was simply based on a sine wave with different phase and offset for each leg, while a similar stiff robot would need complex sensory equipment and an elaborate controller [6].

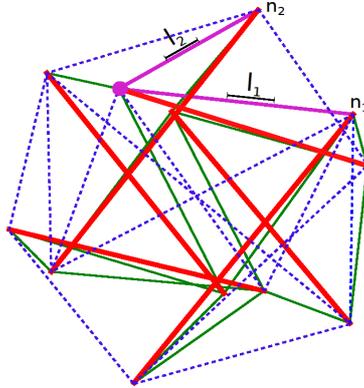


Figure 5.19: End-effector control in a tensegrity robot. In this example we seek to control the lengths l_1 and l_2 of two springs, i.e. the relative position of the endpoint of a bar (large dot) w.r.t two other nodes (n_1 , n_2) of the structure.

In this work, we used tensegrity structures to model compliant systems. However, two important remarks need to be made.

First, the exact dynamics of the system need not be explicitly known to the learning algorithm. This is the underlying idea of reservoir computing: a dynamic system can be used as a computational black box, encoding a non-linearly expanded history of environment interactions in the instantaneous state of the system. Such an abstraction has many advantages, as we can change substrate or construct hybrid systems, while still using the same read-out learning algorithms. It also does not define how the read-out mechanism is actually implemented, and would allow e.g. a neural substrate, electrical wiring or mechanical connections.

Secondly, we can exploit the fact that historically tensegrity structures have been used to model a plethora of complex systems from the micro to the macro scale. Even though tensegrity structures were initially only used in art and architecture, they have now also been successfully applied as a model for cellular cytoskeleton structures [105–107]. At the micro scale, the equations of motions are different and their exact form is often unknown, but we still find compressive elements (e.g. microtubules) and tensile elements (e.g. microfilaments). Inside a single celled organism, there is no central nervous system, but chemical and mechanical interactions define the cell’s behavior, and flagella or cilia allow locomotion [133–136]. Micro-organisms such as nematodes are often capable of rich movement patterns and interaction with the environment while only possessing very simple nervous systems [77, 78]. Based on this, we can hypothesize that the results of our work could provide insight into the fundamental mechanism underlying how simple organisms can perform computations and locomotion required for their survival.

When taking a higher-level viewpoint on the nature of certain aspects of cogni-

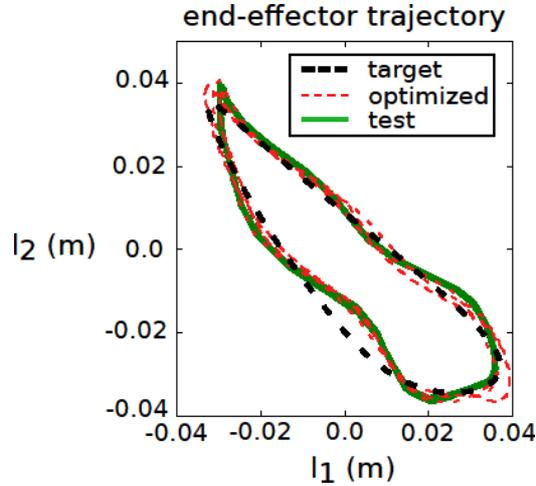


Figure 5.20: Trajectory of the end effector during testing after 75 seconds of training.

tion and computation, our results can offer additional, empirically validated arguments in the quest for understanding cognition in biological organisms. Indeed, we have given several examples of systems in which the computational aspects of locomotion are for the most part physical in nature, making the structures discussed here prime examples of the idea of embodied cognition. Moreover, our analyses allow to quantify the nature of the computation occurring in the substructures, most notably the controller and the physical system. This viewpoint is in our opinion applicable to many of the interactions between the body, sensory inputs and early cognitive layers, but will probably not suffice to fully explain the complete array of cognitive capabilities of human-level intelligence.

When considering cognition as performing computation in the broadest sense, it is clear from our results that this computation is very much divided across the explicit linear control and the implicit non-linear transformations of interactions with the environment, mediated by the physical properties of the structures. Indeed, the idea underlying the principle of Physical Reservoir Computing is precisely that the range of possible dynamical systems which can be used for computation is extremely broad, as are their properties regarding non-linearity or memory. This is not merely a philosophical conjecture: a mathematical framework supporting this claim was recently introduced and proved in [116], showing that any dynamical system of a given size performs the same amount of computation, simply realizing different functions of its external perturbations. We would therefore propose that the question of the true seat of cognitive computation – mental or physical – is rather ill-posed, and that the truth probably lies somewhere in-between. Instead of viewing sensing and cognition as separate but linked entities, we propose that across organisms or even within a single organism, the distinction between mental

cognition and strictly embodied cognition cannot be drawn and likely lies on a continuum.

5.7 Conclusions

In this work, we have introduced an extreme form of embodiment allowing for, so called, Physical Reservoir Computing in a very outspoken sense. It was demonstrated by using highly dynamic and actuated tensegrity structures, effectively computing functions on the history of environment interactions. This allows simple linear learning rules, with a varying degree of reward information, to be able to learn complex locomotion patterns or desired end-effector trajectories.

This provides a number of advantages from a robotic standpoint: the control complexity can be highly reduced, very uninformative reward signals can be used to train complex pattern generators, and the learned control law is robust for perturbations and can easily synchronize with environment interactions.

But from a conceptual point of view, the conclusions are more profound. By demonstrating that dynamic “bodies” only require extremely simple “brains” to implement computations, we effectively opened up a whole spectrum of potential trade-offs between brain-based computation and body-based computation. The powerful computational results from the field of Reservoir Computing [86, 88, 113, 116] can then be used to actually quantify and reason about the computations implemented by the physical body.

Acknowledgments

This research was funded by a Ph.D. fellowship of the Research Foundation - Flanders (FWO) and the European Community’s Seventh Framework Programme FP7/2007-2013 Challenge 2 Cognitive Systems, Interaction, Robotics under grant agreement No 248311 - AMARSi. The authors would like to thank Robert Legenstein for providing the original reward modulated Hebbian learning code from [124].

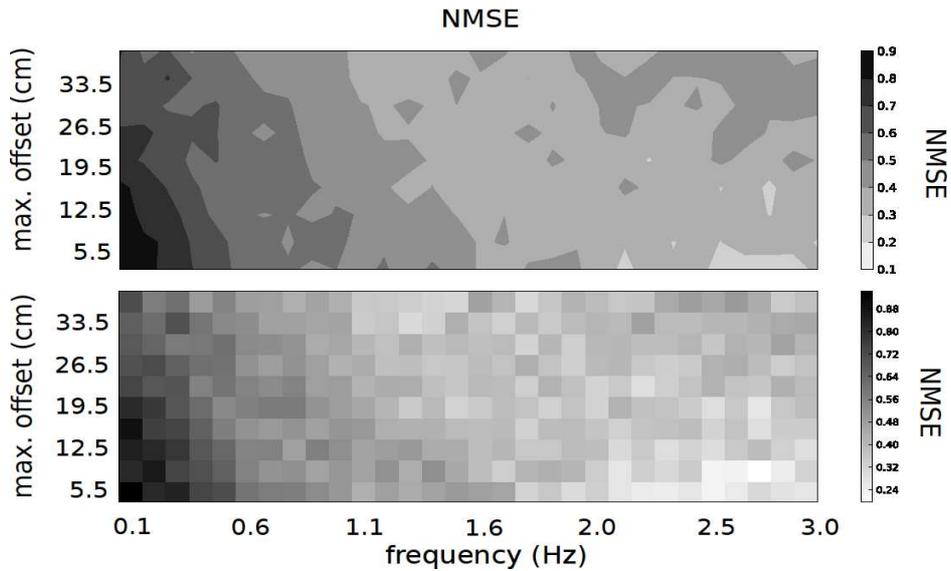


Figure 5.21: Exploiting non-linearity. Plots of the normalized mean-squared error of the first 10 s of testing after training with RLS as a function of the oscillator frequency and the maximum spring equilibrium offset. Top: contour plot showing the different regions. Bottom: result for each combination (frequency, max. offset). The frequency is swept from 0.1 to 3 Hz in steps of 0.1 Hz and the distance from 5.5 cm to 37 cm in steps of 3.5 cm. All tests are performed on the tensegrity icosahedron with random actuated spring connections (between 4 and 8 motors). For each frequency,distance tuple, 50 trials were performed (15000 total), of which the 30 best were retained to reduce the influence of marginal cases. The target was a linear combination of random Matsuoka-type oscillators (12 dimensional). For the task at hand, the system clearly benefits from increasing the frequency of the oscillator and the maximum offset. Very good (computational) results are obtained for a region (bottom right) with only small offset. This region might however not be suited for locomotion applications (limited shape changes).

Chapter 6

Embodiment Enables the Spinal Engine in Quadruped Robot Locomotion

6.1 Introduction

Legged robotics has attracted much more attention from robotic researchers due to its application in rough terrains in nature and in our living environment[1]. Over the past decades, it has been widely accepted that locomotion is generally achieved by the coordination of the legs, and the spine is only considered to be involved in a passive way. Most existing quadruped robots are very similar in their morphology and feature a single rigid body with four legs with individually actuated hips and knees [4, 137–139].

However, the resulting locomotion behavior is much more constrained than its natural counterpart in terms of speed, energy efficiency, maneuverability, and adaptability to rough terrain. From a biological point of view, one of the major differences between robots and animals is the spine. It's function is not only to bear weight, but also to facilitate locomotion. For example, cheetah, the fastest animal on land, exhibits pronounced flexion-extension spinal movement and is able to reach a speed of 110 km/h for a short dash.

Therefore, an alternative biological hypothesis has been proposed by S. A. Gracovetsky that emphasizes on the spinal engine, i.e., locomotion is first achieved by the motion of the spine; the limbs came after, as an improvement but not a substitute[140]. This hypothesis was then extended to quadruped animals featuring flexion-extension and lateral spinal movement[12, 141]. In addition, B. M. Boszczyk demonstrated that mammalian lumbar spines suggest an exact accommodation to specific biomechanical demands, i.e., spinal morphology depends on the locomotion mode in nature[142]. These biological findings indicate that the spine is a crucial element in locomotion.

Inspired by these findings, some robotics researchers came to realize the importance of the spine, but most only focused on the controller of the spinal joint, or regarded the spine as an assistant element to enhance locomotion [7–9]. All the aforementioned studies simply introduced a spinal joint connecting the front and hind part without further study on its morphological parameters. A recent study has demonstrated how spinal morphology affects a passive quadruped robot to go down a slope without external energy and has shed new light on the study of the spine in locomotion [11]. However, more versatile behaviors generated by the spine are expected to emerge and the effect of spinal morphology in locomotion needs to be studied and quantified. We know that this passive bounding behavior is generated through the embodied coupling between the body and the environment, but how the change of spinal morphologies modulates this embodied coupling is still unknown. Such an investigation is challenging because the coupling between controller, body, and environment [10, 143] is usually nonlinear and complex. One promising approach is to take information theoretic measures to quantify information structure in sensorimotor interactions because these measures can detect nonlinear relations between variables [144–149].

In this paper, we introduce a biologically-inspired multiple degree-of-freedom spine model [150, 151] to a real quadruped robot to demonstrate how versatile behaviors (bounding, trotting, and turning) can be generated by the spinal engine through embodied coupling between the controller, the body and the environment. Moreover, three different spinal morphological configurations are introduced to the robot. Then their effects in bounding and trotting gaits are analyzed. We further investigate the effect of these three spinal morphologies on the bounding gait based on information theoretic measures that are able to characterize the sensorimotor interactions. The results suggest that through the embodied interaction with the environment, in particular through sensorimotor coordination, information structure is induced in the sensory data. The correspondence between the observed behaviors of the robot and the information structure is discussed in detail.

6.2 Experimental setting

6.2.1 Quadruped robot

We built a robot, called Kitty, with a flexible actuated spine, to test the concept of the spinal engine based on the interaction between the controller, the body and the environment. An overview of the whole robot, spine structure, spine morphology, controller design, and data acquisition are presented in this section.

Overview of the whole robot

Kitty is equipped with a flexible spine (29 cm wide, 32 cm long, 20 cm high, and 1.4 kg) (Fig. 6.1 (a)). Three linear springs are mounted in each stick-shaped leg

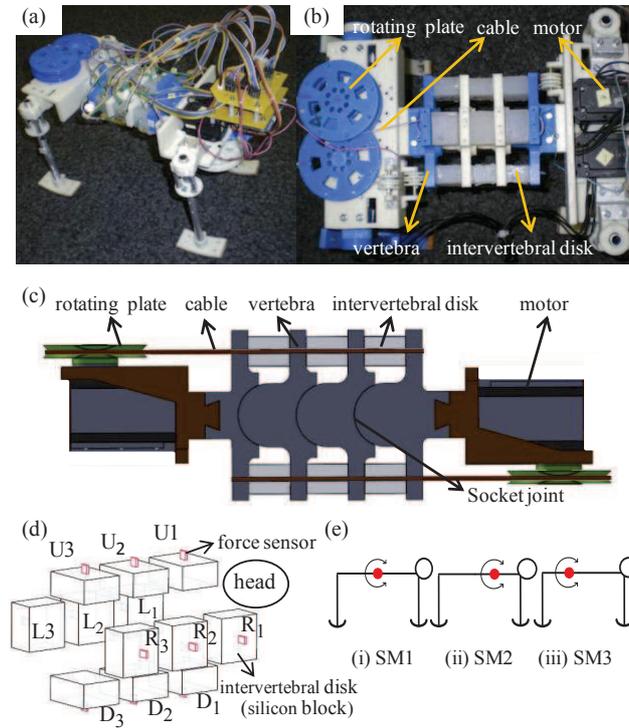


Figure 6.1: (a) Kitty equipped with a tendon-driven spine. (b) A biologically inspired spine. (c) Cross section of the artificial spine: sagittal view. (d) The arrangement of sensors in the spine. Red cubic stands for the force sensor embedded in the silicon block. (e) Schematic representation of robot equipped with an actuated spine whose virtual joint is in the middle (i), front (ii) and the rear (iii) part of the body. The red point stands for the virtual spinal joint.

to cushion the shock from the ground. The legs are fixed to the body and have no relative rotation with respect to it. The bottoms of the feet are glued with asymmetrical friction material to guide the walking direction.

Biologically-inspired spine structure

As an essential structure for both weight bearing and locomotion, the spine is subject to the conflict of providing maximal stability while maintaining crucial mobility. It consists of discrete bony elements, namely vertebrae, joined by passive ligaments and separated by intervertebral discs [152]. The Zygapophyseal joints are dynamically controlled by muscular activation.

Fig. 6.1 (b) shows an artificial spine endowed with biological characteristics. It consists of cross-shaped rigid vertebrae, made of acrylonitrile-butadiene-styrene (ABS) plastic, silicon blocks, and cables driven by motors. As shown in Fig. 6.1

(c), the vertebrae are separated by the silicon blocks, which work as intervertebral discs, and connected by four cables through themselves and the silicon blocks. The four cables are pulled respectively by four electronic motors through the rotating plates. In this design, multiple socket-ball joints are employed to produce a more versatile posture and a wider motion space (Fig. 6.1 (c)). The resulting spine can be bent in all directions within a certain predefined angle and form an asymmetrical complex configuration by rearranging the silicone blocks in between.

Spine morphologies

As we all know, some biological features, e.g., unevenly distributed muscles in the spine, asymmetrical muscle stiffness, and the shape of the spinal column [153], all have the possibility to make the body bend more in one direction and less in another direction. In order to study the effect of asymmetrical structure in locomotion, we design a virtual joint on spine and change its position with the aim of gaining a better understanding of the underlying mechanism of the spine.

We define the virtual spinal joint (VSJ) as a point where the spine is more likely to achieve wider bending movement. The position of the VSJ can be changed by removing some of the silicon blocks from the spine. Twelve silicon blocks are distributed in the spine and three at each side, as shown in Fig. 6.1(d). We name them by the combination of the direction, e.g., up (U), down (D), left (L), right (R) and the order, e.g., 1 (in the front), 2 (in the middle), 3 (in the rear).

We specify the position of the VSJ in the middle, if the silicon blocks fill in all the gaps between the vertebrae and distribute evenly along the spine (Fig. 6.1(e) (i)). The resultant morphology is named spinal morphology 1 (SM1). The morphology, without D1, L1, and R1, which tends to bend more in the front, is called spinal morphology 2 (SM2) (Fig. 6.1(e) (ii)). Spinal morphology 3 (SM3) is defined as the one without D3, L3, and R3 that bends easily in the rear part of the spine (Fig. 6.1(e) (iii)).

Since only two silicon blocks are located in the downside of the spine in SM1 and SM2, their stiffness in this side is less than the one in SM1 which consists of three silicon blocks in the same side. Each side of spine is a complex system, because it is made up of three or two soft silicon blocks working as compression springs, and four rigid plastic segments. Therefore, we simplified the procedure to measure the stiffness in each side of spine as follows. When one side of the spine is pulled, the other three are relaxed. The cable connecting through this side is pulled by a motor to a target position ($\frac{2\pi}{3}$ radian) and the silicon blocks get compressed. We measured the force exerting on this side and its deformation. Next we calculated its stiffness by dividing the measured force by the deformation. We found that stiffness of the side with three silicon blocks is 0.5 (N/mm), while the one with two blocks is 0.33 (N/mm).

Controller design

Four electronic motors (Dynamixel RX 28) are mounted at the front and rear parts of the body to generate force to control the movement of the spine. The motors are serially connected to a PC by a USB2Dynamixel adapter. Position control is taken to drive the spine. The goal position should be set within a valid range from $-\frac{5\pi}{6}$ to $+\frac{5\pi}{6}$, due to the limitation of the motors. Sinusoid functions with tunable parameters are used as control signals for four motors to generate versatile spinal movement (Table 6.1). We represent these motor commands as MU, ML, MR, and MD, according to the spinal parts they actuate, such as up, left, right, and down, respectively.

Each force sensor (FSR400) is inserted into a silicon block to collect force information generated by the compression of the spine (Fig. 6.1 (d)). The naming of the force sensors is the same as the silicon blocks. An Arduino Uno board [154] is used to collect analog data and transfer them to the computer.

6.2.2 Information theoretic measures

As we mentioned in Section I, the behavior of the robot is generated from reciprocal and dynamical coupling between the control, the body, and the environment. This means that any changes in the behavior of the robot can be characterized by changes of this coupling regime. In order to analyze this coupling, simulation models are often constructed to emulate the properties of the physical platform. However, in case of a physical platform that contains soft material, it is difficult to fine-tune a parameter, such as a collision coefficient and elasticity, even if the setting is simple. In such cases, the information theoretic approach can be effectively used to characterize the system in detail. It is intrinsically a model-free approach and can be used for nonlinear time series. Thus, it has been recently used to characterize the coupling regime of robots [144–149].

Usually, an information theoretic approach makes extensive demands on the

Table 6.1: Parameters for different behaviors

Behavior	A_U^1/A_D	A_R/A_L	F_{UD}^2	F_{RL}	ϕ_D^3	ϕ_R	ϕ_L
Bounding	$\frac{2\pi}{3}$	0	0.01	0	π	0	0
Trotting	0	$\frac{2\pi}{3}$	0	0.01	0	π	0
Turning right	$\frac{2\pi}{3}$	$\frac{2\pi}{3}$	0.01	0.01	π	π	0
Turning left	$\frac{2\pi}{3}$	$\frac{2\pi}{3}$	0.01	0.01	π	0	π

¹ $A_U/A_D/A_R/A_L$: amplitude of each side of spine (radian).

² F_{UD}/F_{RL} : frequency of up-down/right-left (hertz).

³ ϕ_R/ϕ_L : phase lag at each side relative to the upside signal (radian).

amount of data and preconditioning of them, such as fine-tuning parameters and data discretization, when applied to a continuous time series. In addition, it is often sensitive to noise, which limits the applications in a real-world physical experiment. In order to overcome this limitation, *permutation entropy* (PE) was proposed [155]. Permutation entropy quantifies the uncertainty of orderings between values, unlike the usual entropy which quantifies that of values themselves. Despite the differences between the procedures, it was proved that the PE rate is equal to the usual entropy rate for any finite-state stationary stochastic process [156, 157]. The method is especially useful since it contains a natural symbolization procedure of values, which usually requires preprocessing to determine the appropriate bin size for the discretization of the acquired data. It was also shown that PE is robust to noise, which is common to real-world time series [155].

Let x_t represent a set of amplitude values of time series x'_t with a given embedding dimension m , and \hat{x}_t as corresponding symbols. Based on the permutations of the values, \hat{x}_t is generated as follows: $x_t = \{x'_t, x'_{t+1}, \dots, x'_{t+(m-1)}\}$, which are arranged in ascending order, $\{x'_{t+(k_{t1}-1)} \leq x'_{t+(k_{t2}-1)} \leq \dots \leq x'_{t+(k_{tm}-1)}\}$. A symbol is thus defined as $\hat{x}_t \equiv (k_{t1}, k_{t2}, \dots, k_{tm}) \in X_t$. (Actually, for computational efficiency, we used the procedure introduced in [156, 157], which uses rank sequences. In [157], the use of the rank sequences is proved to be equal to the one presented here. It is useful to uniquely map x_t onto $m!$ possible permutations, including the case of equal amplitude values.) Based on the generated symbol \hat{x}_t , PE ($H(X_t)$) is defined as

$$H(X_t) = - \sum_{\hat{x}_t} p(\hat{x}_t) \log p(\hat{x}_t), \quad (6.1)$$

where $p(\hat{x}_t)$ is a single probability associated with \hat{x}_t . In this paper, based on PE , we adopt two information theoretic measures and in each case, $m=6$ throughout this paper.

Permutation mutual information

Mutual information is a general measure of association between two or more random variables, naturally encompassing both linear and nonlinear dependencies [158]. We also consider the permutation version of mutual information in this paper. Permutation mutual information (PMI) measures statistical independence as:

$$PMI(X_t, Y_t) = \sum_{\hat{x}_t, \hat{y}_t} p(\hat{x}_t, \hat{y}_t) \log \frac{p(\hat{x}_t, \hat{y}_t)}{p(\hat{x}_t)p(\hat{y}_t)}, \quad (6.2)$$

where $p(\hat{x}_t, \hat{y}_t)$ is a joint probability associated with states \hat{x}_t and \hat{y}_t . For statistically independent distributions, $p(\hat{x}_t, \hat{y}_t) = p(\hat{x}_t)p(\hat{y}_t)$ and $PMI(X_t, Y_t) = 0$. If there exist statistical dependencies, $PMI(X_t, Y_t) > 0$.

Symbolic transfer entropy

Transfer entropy (TE) is a measure of the information transfer from the driving system (Y) to the responding system (X) [159]. The permutation version of transfer entropy is proposed in [160] and is called *Symbolic Transfer Entropy (STE)*, expressed as:

$$STE(Y \rightarrow X) = \sum_{\hat{x}_{t+1}, \hat{x}_t, \hat{y}_t} p(\hat{x}_{t+1}, \hat{x}_t, \hat{y}_t) \log \frac{p(\hat{x}_{t+1} | \hat{x}_t, \hat{y}_t)}{p(\hat{x}_{t+1} | \hat{x}_t)}, \quad (6.3)$$

where the index $STE(Y \rightarrow X)$ indicates the influence of \hat{y}_t on \hat{x}_{t+1} and can thus be used to detect the directed information transfer from Y to X . In other words, STE somehow measures how well we can predict the state transition of the system X by knowing a state of system Y . STE is non-negative; any information transfer between the two variables results in $STE > 0$. If the state of \hat{y}_t has no influence on the transition probabilities from \hat{x}_t to \hat{x}_{t+1} , or if the two time series are completely synchronized, $STE = 0$. Recently, TE and STE have been proved to be equivalent if they are considered as the rates within the realm of finite-alphabet stationary ergodic Markov processes [161] and hidden Markov models with ergodic internal processes [162].

6.3 Experiments

Two sets of experiments were conducted. The first one was with SM1. The goal is to show the versatility of spine-generated behaviors (bounding, trotting, and turning). The second set studied the effect of three spinal morphologies (SM1, SM2, and SM3) on the bounding and trotting gaits.

During all the experiments, the average speed and significant difference were calculated based on three trials for each experiment, and one trial consists of 1200 time steps. The unit representing "time" in this paper is one sensing and actuation loop of the control program. One cycle of the motor command is 100 time steps and 12 cycles were run in each trial.

6.3.1 Versatile behaviors

Bounding/Flexion-extension spinal movement

The bounding gait is generated by spinal flexion-extension movements, which are achieved by pulling cables located at the upside and downside of the spine in an alternate way. The side cables are kept the natural length without stretching and relaxing. Fig. 6.2(a) shows the dynamics of the middle spine during one cycle of the bounding gait. The value in the vertical axis in Fig. 6.2 is a normalized sensory output voltage, which represents the force applied to the sensor. The applied force F (Newton) can be computed as $F = \frac{10y}{256-y}$, where y is the value of the vertical

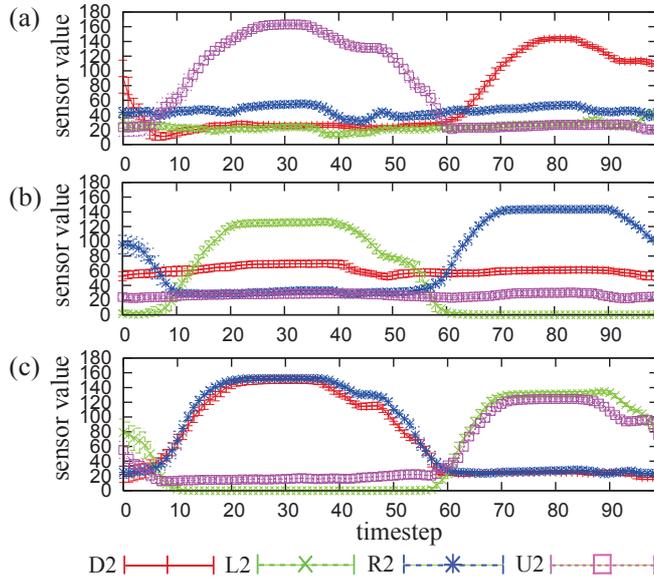


Figure 6.2: Middle spine dynamics based on SM1 during one cycle in the experiments of (a) bounding, (b) trotting, and (c) turning right. Responses of the sensory value of 36 cycles of oscillatory motor commands are used to obtain the average. The error bars represent the standard deviation.

axis. We use y to express the sensory response throughout this paper. We can see that the up and down sensors (red and purple lines in Fig. 6.2 (a)) are compressed alternately as a result of motor commands.

Fig. 6.3(a) shows symmetrical, periodic flexion-extension spinal movements generating power for locomotion. The robot is able to move forward at a speed of 5.76 cm/s with a standard deviation of 0.09 (Fig. 6.4(a)). However, as can be seen from Fig. 6.3(a), its feet slide on the ground due to the lack of ground clearance.

Trotting/Lateral bending

The trotting gait is generated by the spinal lateral bending movement, which is produced by pulling side cables alternately while keeping the up and down cables the natural length. Fig. 6.2(b) shows the middle spine internal dynamics induced by the motor pattern. The spine exhibits symmetrical and periodic lateral bending movement and generates the trotting gait (Fig. 6.3(b)). The robot is able to trot at a speed of 1.7 cm/s with a standard deviation of 0.1 (Fig. 6.4(b)). We observed that the speed can reach 3.0 cm/s by increasing the frequency to 0.025 Hz.

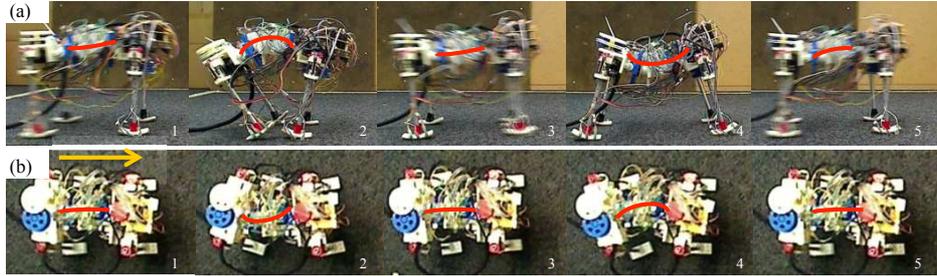


Figure 6.3: Sequential pictures of spinal movement in (a) the bounding gait and (b) the trotting gait. Yellow arrow represents the walking direction.

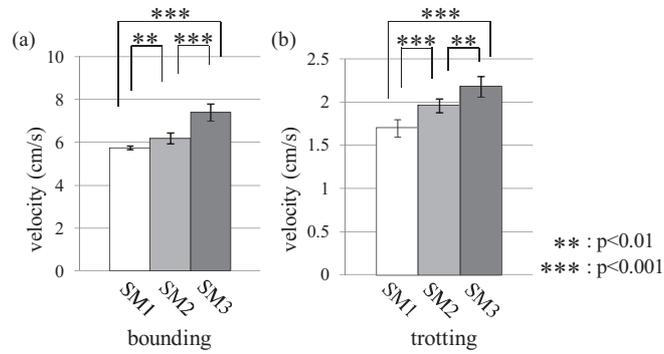


Figure 6.4: Comparisons of the averaged velocity for SM1, SM2, and SM3 for the bounding gait (a) and the trotting gait (b). For both (a) and (b), the error bars show the standard deviation. In (a), the averaged velocity for SM1, SM2, and SM3 are 5.76 ± 0.09 , 6.20 ± 0.26 , and 7.41 ± 0.40 , respectively. In (b), the averaged velocity for SM1, SM2, and SM3 are 1.70 ± 0.1 , 1.96 ± 0.08 , and 2.18 ± 0.12 , respectively. For each plot, asterisks indicate significant differences, **: $p < 0.01$, and ***: $p < 0.001$.

Turning

The turning behavior emerges if the lateral spinal movement is introduced to the spine while flexion-extension spinal movement is retained. The only difference between turning right and left is the flip between ϕ_R and ϕ_L , which are the phase lags with respect to the upside of the spine. Fig. 6.2(c) shows the middle spine dynamics during turning behavior, which corresponds to the motor pattern. The robot is able to turn at a radius of 0.42 m with a standard deviation of 0.02.

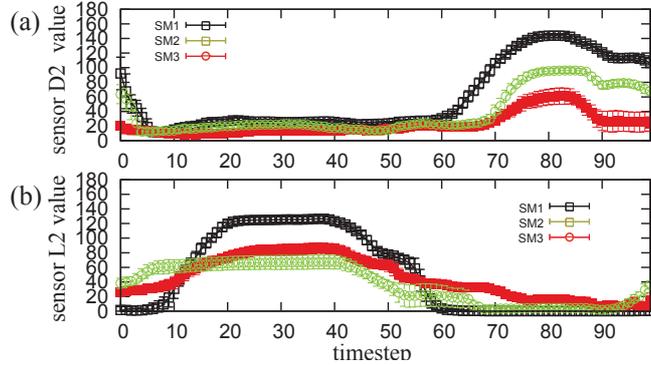


Figure 6.5: Comparison of the typical average response curves in the middle spine. (a) Force sensor D2 is taken as the typical case for the bounding gait, and (b) Force sensor L2 is taken as the typical case for the trotting gait. Responses of the sensory value to 36 cycles of oscillatory motor commands are used to obtain the average. The error bars represent the standard deviation.

6.3.2 The effect of spinal morphology on the bounding and trotting gaits

To capture the internal spine dynamics, we picked force sensors D2 and L2 as typical cases for the bounding and trotting gaits, respectively, due to their sensitiveness to each gait. In both cases, SM1 endures more pressure compared to the rest (Fig. 6.5), because all the silicon blocks fill in the gap and do not offer much freedom. The absence of some silicon blocks from SM2 and SM3 results in less pressure and a wider space to bend, resulting in a higher speed.

The robot with SM3 shows the best performance and is able to reach a speed of 7.41 cm/s with a standard deviation of 0.4 for the bounding gait and a speed of 2.18 cm/s with a standard deviation of 0.12 for the trotting gait (Fig. 6.4). It is easier for SM3 to pull the back legs forward to increase the stride due to the rear spinal joint. SM2 has more freedom in the front part, which is expected to lift up the front legs [151] to increase stride length. However, the circuit boards are put on the head and increase weight, so the force needed to lift up the body is compensated by the weight of circuit boards. Therefore, no ground clearance can be observed. SM1 is the worst case due to the limited bending space.

6.3.3 Information theoretical analysis on the bounding gait

As observed in the previous sections, we can obtain various types of locomotion style only by actuating the spine part. In addition, we confirmed that, especially for the bounding and trotting gaits, even if the motor commands are the same,

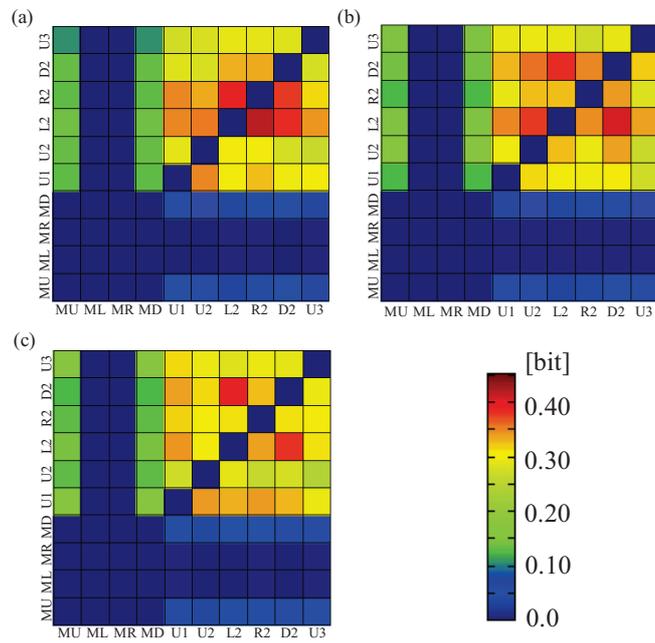


Figure 6.6: Comparisons of the averaged value of STE in the bounding gait condition for (a) SM1, (b) SM2, and (c) SM3. For each morphology, the robot is driven in on-the-ground condition for 3 trials with 1200 time steps each. For each STE , the value is averaged over 3 trials. The information transfers from the variables in horizontal axis to vertical axis.

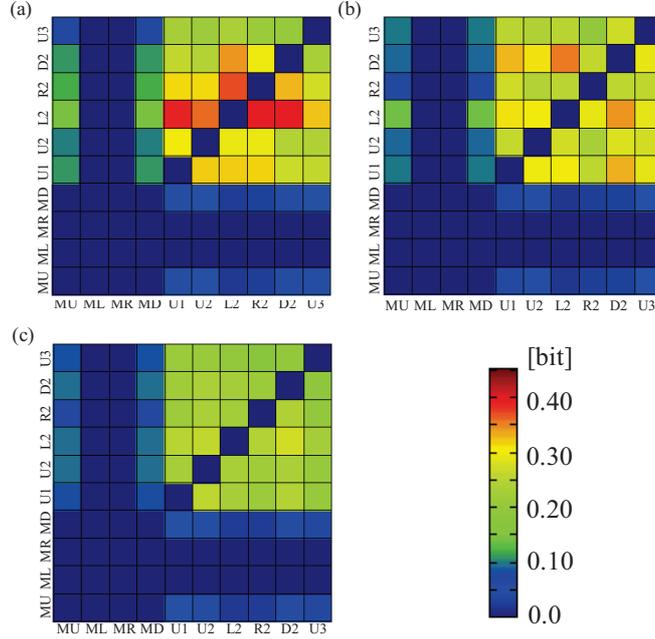


Figure 6.7: Comparisons of the averaged value of STE_{off} in bounding gait condition for (a) SM1, (b) SM2, and (c) SM3. For each morphology, the robot is driven in off-the-ground condition for 3 trials with 1200 time steps each. For each STE_{off} , the value is averaged over 3 trials. The information transfers from the variables in horizontal axis to vertical axis.

only by changing the morphology of the spine, the velocity of the robot changes significantly. As we explained, a behavior of robots is generated by the coupling between the controller, the body, and the environment. Thus, the observed difference in velocity would be caused by the modulation of these couplings induced by the change of spinal morphology. In this section, by focusing on the bounding gait, we aim to characterize how the change of spinal morphology modulates these couplings by using information theoretic measures.

Fig. 6.6 shows the averaged STE between the variables in the bounding gait for SM1, SM2, and SM3. Here, in order to make each morphological condition comparable, we selected sensors that remain invariant in each morphology, namely, U1, U2, L2, R2, D2, and U3, and these spinal variables (SP) are used to characterize the spinal dynamics throughout the analysis. As can be seen from Fig. 6.6, the information transfers from MU and MD to spinal variables, and from spinal variables to spinal variables themselves show remarkably high value in each morphology. These structures are considered to be reflecting all the couplings between the controller, the body, and the environment.

Next, we focus on the relationship only between the controller and the body in each morphology. To check this, we run experiments for each morphology by

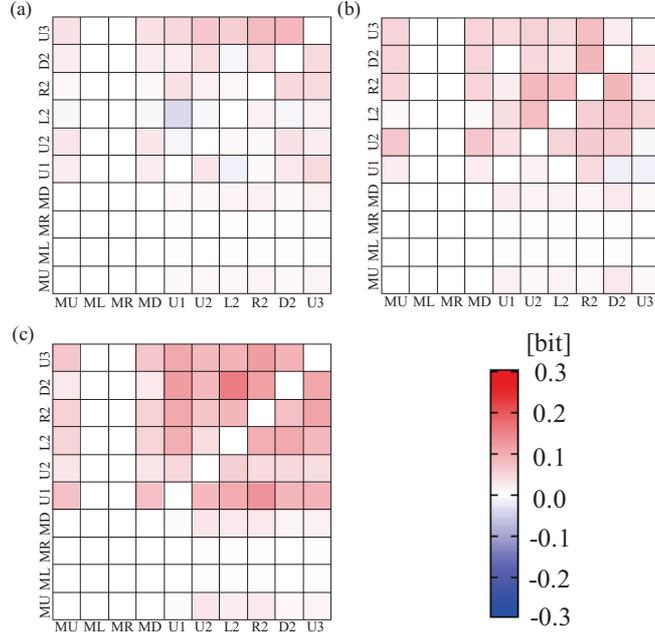


Figure 6.8: Comparisons of the averaged value of $STE_{subtract}$ in the bounding gait condition for (a) SM1, (b) SM2, and (c) SM3. In order to calculate $STE_{subtract}$, we first calculated STE_{off} by using 3600 time steps in the off-the-ground condition for each morphology. Then, we subtracted this STE_{off} from STE in each trial and obtained the averaged $STE_{subtract}$. Each plot shows the subtracted information transfer from the variables in horizontal axis to vertical axis.

lifting the robot up, so the robot's legs cannot touch the ground (off-the-ground case). Other conditions, such as sensorimotor conditions, are kept the same. Fig. 6.7 shows the results of the averaged STE between variables in the off-the-ground case (we call STE in the off-the-ground case STE_{off}). As can be seen from Fig. 6.7, the values seem to be lower than those in Fig. 6.6 in general. This result suggests that the interaction with the environment is enhancing the information transfer between the variables. In order to confirm this point clearly, we subtracted STE_{off} from STE (we call this $STE_{subtract}$) and checked how much information is additionally transferred by the interaction with the environment. Fig. 6.8 shows the results of the averaged value of $STE_{subtract}$. We can confirm that, according to the morphological change from SM1, to SM2, and to SM3, there exists a tendency for $STE_{subtract}$ to increase.

We checked this increase of information transfer further by summing up the information transfer from M to SP and from SP to SP, and subtracting the corresponding STE_{off} , which are respectively denoted as $STE_{subtract,total}(M \rightarrow SP)$

and $STE_{subtract,total}(SP \rightarrow SP)$, and expressed as follows:

$$STE_{subtract,total}(M \rightarrow SP) = \sum_{i=1}^4 \sum_{j=1}^6 (STE(M_i \rightarrow SP_j) - STE_{off}(M_i \rightarrow SP_j)), \quad (6.4)$$

$$STE_{subtract,total}(SP \rightarrow SP) = \sum_{i=1}^6 \sum_{j=1}^6 (STE(SP_i \rightarrow SP_j) - STE_{off}(SP_i \rightarrow SP_j)), \quad (6.5)$$

where $M_{1\sim 4}$ correspond to MU, ML, MR, and MD, respectively. Also, $SP_{1\sim 6}$ correspond to U1, U2, L2, R2, D2, and U3, respectively. In addition, we calculated the increase of associations between the variables by summing up the subtracted PMI in a similar manner with $STE_{subtract,total}$ between M and SP ($PMI_{subtract,total}(M, SP)$) and between SP and SP ($PMI_{subtract,total}(SP, SP)$). Note that, since PMI has no directionality, we avoided summing up the case of the same pair of the same variables.

Comparisons of the averaged values of each measure are shown in Fig. 6.9. We can confirm that the degree of increase of the information transfer and the associations between the variables tends to be enhanced according to the change of spinal morphology in the order of SM1, SM2, and SM3. Interestingly, this result seems to be corresponding to the change of velocity according to the morphological change. This may suggest that the enhancement of the information transfer and associations brought about by the interaction between the environment in the relation around M and SP leads to the increase of velocity.

6.4 Conclusion and discussion

We demonstrated versatile spine-driven behaviors through embodied coupling between the controller, the body, and the environment. This spine plays a dominant role in generating movement and transferring energy to the legs, which are able to achieve versatile behaviors. This quadruped robot exhibits pronounced flexion-extension and lateral spinal movements, which result in the bounding and the trotting gaits, respectively. Turning behavior can also be realized by the combination of the bounding and trotting gaits. More spinal morphologies have been explored in terms of the VSJ based on the bounding and trotting gaits. Experimental results showed that the speed of SM3 and SM2 is greater than SM1 due to the freedom offered by the absence of some silicon blocks in the spine. SM3 with a rear VSJ showed the best performance in both cases in terms of speed, because a rear VSJ helps rear part of body to gain more freedom to bend, pull the rear legs forward, thus leading to a longer stride. However, there is a trade-off between stability and speed [151]. If more silicon blocks are removed, then more freedom and greater bending angles can be achieved. This results in less stability. In addition,

we observed that the robot's performance is better if the stiffness in the downside of the spine is lower than the one in the upside. This finding is consistent with simulation studies which state that the spine stiffness in extension is greater than that in flexion [163].

As we all know, in the biological system, a spine consists of multiple segments, which makes it difficult to model and analyze. However, our study suggests that information theoretic measures might be useful to quantify how the spine affects locomotion through the interaction with the environment. By using the information theoretic analysis, we characterized the information structure of the sensorimotor variables in the bounding gait with different spinal morphologies. As a result, we show that the information transfers and the associations between the variables brought about by the interaction with the environment tend to increase according to the change of spinal morphology. This enhancement of the information structure seems to have a correspondence with the change of velocity. This correspondence should be further investigated in future work.

In this study, the simplest legs without actuation were introduced to investigate spine-driven locomotion. We observed that the legs slide on the ground most of the time due to the lack of ground clearance. This problem can be solved by adding leg actuation, as explained by Gracovetsky, who states that the role of legs is to achieve ground clearance and overcome obstacles, whereas the function of the spine is to generate main force and movement for locomotion [141]. In future, new actuated legs will be designed and applied to this robot to increase dynamical performance. We will also install touch sensors and joint sensors to the legs to sense their internal dynamics to gain a better understanding of the underlying mechanism of the function of the spine in locomotion, e.g., how the force propagates from the spine to the legs based on the information theoretical method.

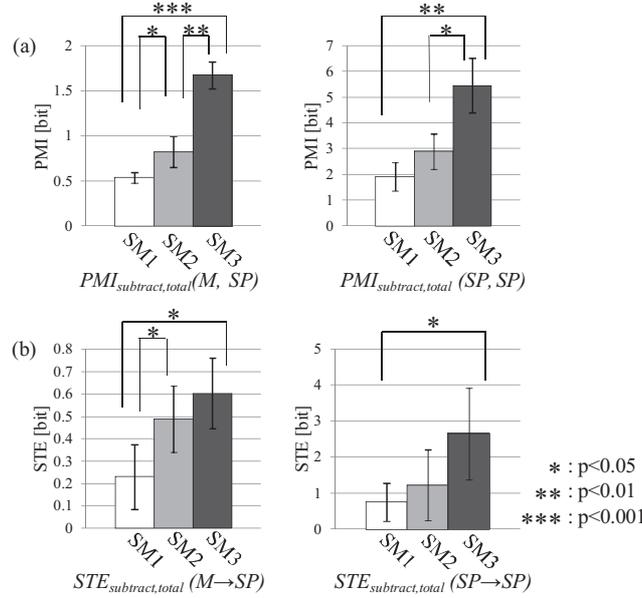


Figure 6.9: (a) Comparisons of $PMI_{subtract,total}(M, SP)$ (left) and $PMI_{subtract,total}(SP, SP)$ (right) according to the morphological changes of the spine. The averaged values of $PMI_{subtract,total}(M, SP)$ in SM1, SM2, and SM3 are 0.54 ± 0.06 , 0.82 ± 0.17 , and 1.67 ± 0.15 , respectively. The averaged values of $PMI_{subtract,total}(SP, SP)$ in SM1, SM2, and SM3 are 1.91 ± 0.55 , 2.89 ± 0.69 , and 5.46 ± 1.06 , respectively. (b) Comparisons of $STE_{subtract,total}(M \rightarrow SP)$ (left) and $STE_{subtract,total}(SP \rightarrow SP)$ (right) according to the morphological changes of the spine. The averaged values of $STE_{subtract,total}(M \rightarrow SP)$ in SM1, SM2, and SM3 are 0.23 ± 0.14 , 0.49 ± 0.15 , and 0.60 ± 0.16 , respectively. The averaged values of $STE_{subtract,total}(SP \rightarrow SP)$ in SM1, SM2, and SM3 are 0.75 ± 0.53 , 1.23 ± 0.98 , and 2.65 ± 1.27 , respectively. In order to calculate the averaged value of each measures, we used the same procedure of subtractions explained in Fig. 6.8. For all the plots, the error bars show the standard deviation, and asterisks indicate significant difference, *: $p < 0.05$, **: $p < 0.01$, and ***: $p < 0.001$.

Appendix A

Supplement material of Chapter 2

Angle of attack estimation from empirical data

In the experimental data of reference [51] the angle of the right limb is measure against the vertical. We use this information to estimate the angle of the leg at landing based in two facts. First, the angle of the leg changes more its velocity in the swing phase (the foot is not in contact with ground) than in the support phase (the foot is in contact with the ground), and second, as soon as the leg changes from the swing phase to the support phase there is a big change of the angular velocity due to the impact of the food against the ground when it lands.

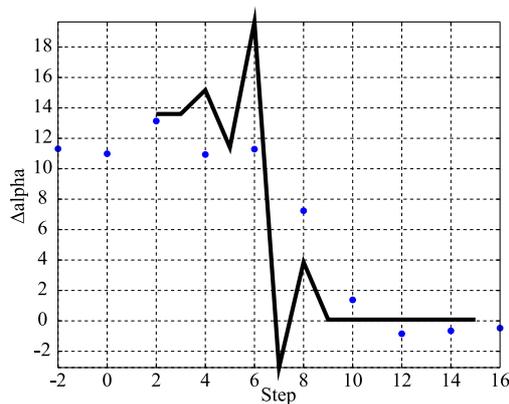


Figure A.1: (Color online) Change of the angle of attack in the running to walking transition. The solid line represent the change of the angle of attack in the model and the dotted line represent the change of the angle of attack in a human experiment. In both case there is a transition from running to walking.

The angle of attack identified using this conditions allow the comparison of the strategy in human locomotion and the proposed model. The model qualitatively develops a similar strategy. The difference of the angle of attack between the steady state gait (e.g. walking or running) from the experiment and the model is around five degrees. To facilitate the qualitative comparison of the angle of attack, we evaluate the change of the angle of attack against the angle of attack of walking. Using this measurement, we can avoid the difference of five degrees and focus in the strategy for gait transition.

Fig. A.1 and Fig. A.2 show that the strategy developed with the model has similar steps and matches the change of the angle of attack in the transition. Fig. A.1 shows a more drastic change of the angles of attack compare with the experiment result, however the data of the experiment is from one leg which allow the identification of the angle of attack every two steps. This can be emulated with the model selecting only the even or the odd steps. In any of these cases, the change of the angles of attack is going to look less drastic and qualitatively more similar to the ones from the experiment.

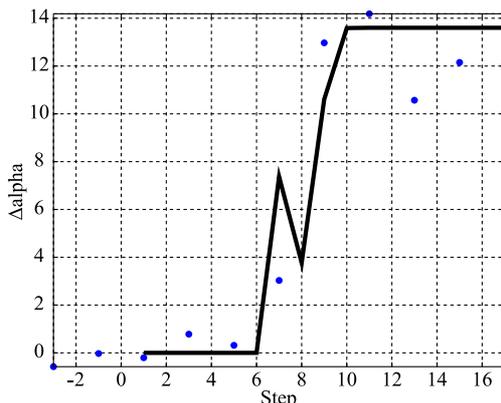


Figure A.2: (Color online) Change of the angle of attack against in the walking to running transition. The solid line represent the change of the angle of attack in the model and the dotted line represent the angle of attack in a human experiment. In both case there is a transition from walking to running.

Change of phase of hip excursion before and after transition

As shown in Figure A.3 (left axis), during walking and running the hip follows and oscillatory trajectory over time. We compare the phase of these oscillations with

Strategy	$\mathcal{W} \rightarrow \mathcal{R}$	$\mathcal{R} \rightarrow \mathcal{W}$
Const. Froude number	36.3°	35.3°
Const. hip excursion	55.3°	51.5°
Fitting experiment	109.0°	110.9°
Experiment	-35.0°	86.8°

Table A.1: Change of phases for three strategies and experimental data. None of the transitions shows a phase change in full accordance with the experimental data. The absolute value of the phase change for the transition from walking to running at constant Froude number is very close to the experimental value, however the direction of the change is opposite.

respect to the moment of transition. The moment of transition was identified as follows:

1. Calculate the analytic signal of the hip trajectory by means of the Hilbert transform, e.g. `hilbert` function in GNU Octave’s signal package [49].
2. Obtain the phase of the signal from the angle of the analytic signal.
3. Take the time derivative of the phase, this is an approximation of the frequency of the oscillations as a function of time.
4. Search for the highest peak in the frequency signal. This point separates the regions of walking from the regions of running.

Figure A.3 shows the frequency signal superimposed to the experimental data. The transition point is indicated with a vertical arrow. Taking this point as the origin of time, we calculate the initial phase of walking and the initial phase of running, by means of fitting a first order polynomial to the phase signal of each gait. This is shown in Figure A.4 when applied to the experimental data. The change of phase is calculated as the difference of these initial phases normalized to the interval $(-\pi, \pi]$. The exact same analysis was applied to all the signals, simulated and experimental.

The changes of phase for the three transition strategies presented in the paper are summarized in Table A.1. All the simulated examples are able to match the direction of the change of phase in the running to walking transition. However, none of the transitions shows a phase change in full accordance with the experimental data. The absolute value of the phase change for the transition from walking to running at constant Froude number is very close to the experimental value, however the direction of the change is opposite.

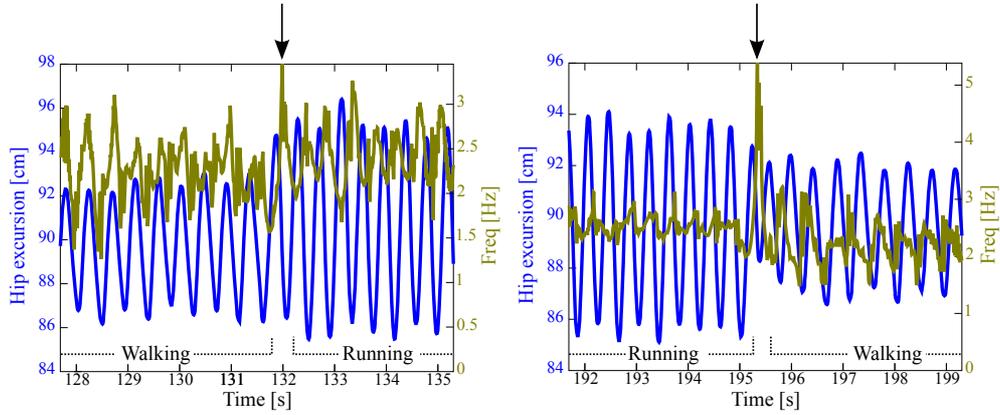


Figure A.3: (Color online) Transition point determination. Plot of the experimental data (left axis) and the the derivative of the phase signal (right axis). this derivative gives a frequency signal that presents a peak during the transition that is used to determine the transition point (vertical arrow).

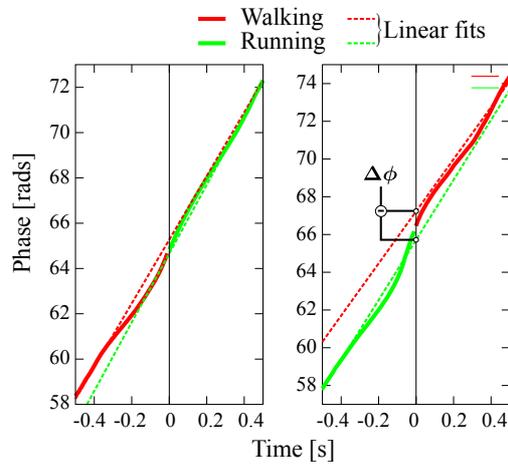


Figure A.4: (Color online) Phase difference calculation. Taking the point of transition as the origin of time, the phase difference is calculate from the intercept of linear fits applied to the two parts of the phase signal. Solid lines show the phase signal for walking and running. Dashed lines show the linear fits.

Appendix B

Supplement material of Chapter 5

Reducing communication load

The GD rule is nothing more than a simplification of the RLS approach to overcome updating a covariance matrix, at the cost of slower convergence and an additional parameter (the learning rate). This can be easily seen by replacing \mathbf{P}^{rls} with the identity matrix. We then have:

$$\mathbf{W}(t) = \mathbf{W}(t - \Delta t) - e \frac{\mathbf{x}^T(t)}{\|\mathbf{x}(t)\|^2}, \quad (\text{B.1})$$

which is exactly the GD rule with a varying learning rate of $\|\mathbf{x}(t)\|^{-2}$.

As such the GD rule is almost equivalent to the RLS rule for uncorrelated input variables. This also indicates how RLS can be implemented in a biologically plausible way. By adding a (linear) layer to project the inputs on their principal components, one obtains uncorrelated input variables. The Generalized Hebbian Algorithm (GHA) can be used to implement this [164]. So as long as an error signal is available for each motor signal, both RLS and GD are viable options which are straightforward to implement.

The GHA is an extension of Oja's rule [165] to multiple output networks. In fact, it can be seen as a stacked version of Oja's rule. From each additional output, all previous principal components are extracted. Oja's rule itself is simply a stabilized version of the classic Hebbian learning rule and it is obtained after doing a first order Taylor approximation of the Hebbian rule with normalization to keep norm of the weights equal to one.

In matrix form the GHA algorithm can be written as:

$$\mathbf{y}_{gha}(t) = \mathbf{W}_{gha}(t - \Delta t)\mathbf{x}(t) \quad (\text{B.2})$$

$$\mathbf{W}_{gha}(t) = \mathbf{W}_{gha}(t - \Delta t) + \eta_{gha}(\mathbf{y}_{gha}(t)\mathbf{x}^T(t) - \text{LT}[\mathbf{y}_{gha}(t)\mathbf{y}_{gha}^T(t)]\mathbf{W}_{gha}(t - \Delta t)) \quad (\text{B.3})$$

where $\text{LT}(\mathbf{X})$ sets all elements on or above the diagonal to zero, hence it removes the (estimated) principal components from the previous outputs.

One application of the additional decorrelating layer, is that it can be used to reduce the number of signals to be communicated. Indeed, the first principal components will extract the most fundamental properties of the signals in the system. Hence, it provides a natural way of forcing the RMH/EH algorithm to focus on the first principal components. This is due to two reasons. First, the first principal components will typically contain *simpler* signals. Secondly, the first principal components are more stable, hence they will tend to be reinforced more than fluctuating inputs.

Fig. B.1 shows information on the GHA layer when used in combination with RMH (in the loop). Shown are the correlation matrices of the raw sensor data (spring forces and their derivatives), the input to the GHA layer (10 time steps delayed raw sensor data) and the correlation of the GHA output trained with 100 outputs. Although the GHA output is not completely decorrelated (this can be improved by increasing the training time), the correlation between variables is much less pronounced than before.

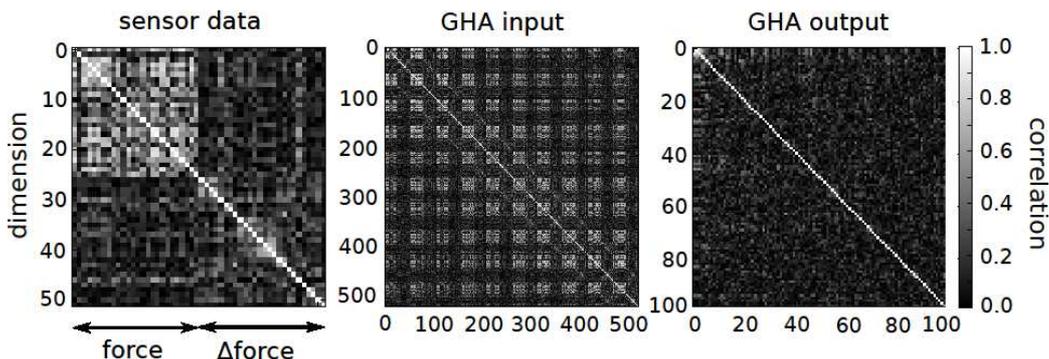


Figure B.1: Correlation of the input data without and with the decorrelating GHA layer. Left: correlation of the inputs after the GHA layer. 100 outputs were trained. Center: correlation of the sensor data (spring forces and derivatives of spring forces). Right: correlation of the original input data with 10 delay lines.

Fig. B.2 shows the weights distribution after training with GHA combined with RMH. The same attractor from Fig. 5.11 was trained, with a similar result. The weights from the first principal components have a larger magnitude.

Simulation details

We simulated the tensegrity systems by vectorizing the DAE description (see Section 5.2). The VODE solver from [166] was used with a variable time step. Ground

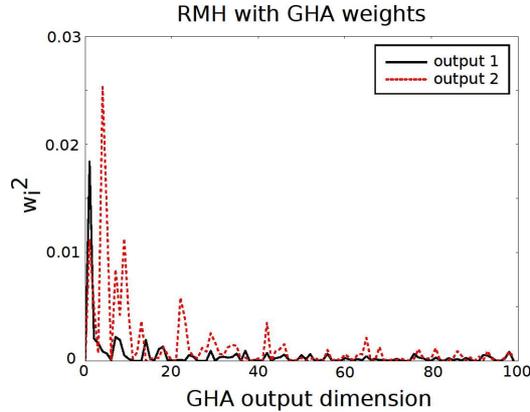


Figure B.2: A possible application of the GHA layer: reducing the communication needs. Shown is the weight distribution of the two outputs as in Fig. 5.11, but this time combined with GHA. Because the first principal components are more stable and extract the lowest frequency signals, the RMH algorithm will tend to assign larger weights to them.

collisions were modeled as external forces acting on the endpoints of the rods. The reaction forces were computed as in [167]. Internal (bar-bar) collisions were not taken into account.

We assumed ideal motors (instant change of spring equilibrium length), but limited their range and velocity. More precisely, each motor could change the spring length by at most 0.3 m at 0.3m/s. For the tensegrity icosahedron we normally actuated non-structural springs (i.e. the structure remains stable without these springs), hence the motors need not be actuated at rest. The actuator springs (for the icosahedron) had an equilibrium length equal to the distance between their endpoints with the actuator springs removed. The magnitude of the force on the springs was usually $< 10N$. We assumed that the rods weighed 0.4 kg/m. The rods had lengths between 0.2 m and 1 m depending on the configuration (around 0.5 m for the icosahedron, which had a mass of about 1.4 kg).

That these are reasonable conditions, can be seen by computing the required motor power. Assuming a constant speed of 0.3 m/s with an applied load of 10 N, 3 Watts of motor power is needed. Lightweight (< 50 g) DC motors are available in this range. Small lithium polymer batteries can deliver enough power and energy to drive such motors over longer periods of time.

A controller frequency of 50 Hz was used, to prevent stringent communication requirements. We can estimate the total bandwidth required for e.g. the tensegrity icosahedron. Spring forces can be measured using strain gages. ADCs in this frequency range are commonly available at low price with a precision up to 24 bits (3 bytes). The icosahedron has 24 springs in its minimal configuration and 60 when fully connected. Hence the minimal communication bandwidth is: (2 x

springs x 3 bytes/spring + overhead)* 50 Hz (spring force and derivative). We assume broadcasting is used, as the algorithms can be implemented locally on each rod. The overhead can contain e.g. the target signal if required. Let us assume an overhead of 5 bytes per motor: (2 x springs x 3 bytes/spring + motor x 5 bytes/motor)* 50 Hz. Hence for the icosahedron in its minimal configuration with 10 actuators (the forces on the actuator springs are not used), we obtain a bandwidth of only 9.7 KB/s.

Feedback linearizability of a single bar

For the sake of completeness, we investigate the feedback linearizability of a bar attached to springs. Consider a single bar attached to 3 springs at each outer end. This is a minimal assumption on freestanding tensegrity structures (each endpoint needs at least 3 attached springs, or there will be a mechanism). Each of the springs is fixed to the rod at one end and to a fixed location at the other.

The force on an endpoint due to a spring is given by:

$$\mathbf{f} = k\left(1 - \frac{l_0}{\|\mathbf{n} - \mathbf{p}\|}\right)(\mathbf{n} - \mathbf{p}) = u(\mathbf{n} - \mathbf{p}), \quad (\text{B.4})$$

where we have assumed that the springs are always in tension. Here u is the transformed control input. Assuming ideal motors, such an input always exists when the springs are strictly in tension. Note that in practice, this can easily be implemented as the spring length can be measured by a force sensor on the spring.

The force on an endpoint is simply the sum of the forces due to the 3 attached springs:

$$\mathbf{f}_n = \sum_{i=0}^2 u_i(\mathbf{n} - \mathbf{p}_i). \quad (\text{B.5})$$

Assuming u_i to be unrestricted, the column space of the matrix $\begin{bmatrix} \mathbf{n} - \mathbf{p}_0 & \mathbf{n} - \mathbf{p}_1 & \mathbf{n} - \mathbf{p}_2 \end{bmatrix}$ is \mathbb{R}^3 . This is trivial: as long as \mathbf{n} does not lie on the plane defined by the endpoints of the springs, this is fulfilled.

We will again assume the bar to be infinitely thin and the mass to be evenly distributed along the bar and follow the description from [98]. Now define a minimal set of generalized coordinates:

$$\mathbf{q} = \begin{bmatrix} \mathbf{r} \\ \boldsymbol{\beta} \end{bmatrix}, \quad (\text{B.6})$$

where \mathbf{r} is the center of mass of the bar and $\boldsymbol{\beta} = \begin{bmatrix} \theta \\ \phi \end{bmatrix}$, the orientation of the bar (around two orthogonal axes).

Now we define the matrix \mathbf{K} :

$$\mathbf{K} = \begin{bmatrix} \sin(\phi)^2 & 0 \\ 0 & 1 \end{bmatrix}. \quad (\text{B.7})$$

This matrix is singular for $\phi \in \{0, k\pi\}$.

Now define the vector $\mathbf{g}(\mathbf{q}, \dot{\mathbf{q}})$:

$$\mathbf{g}(\mathbf{q}, \dot{\mathbf{q}}) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ -2j \sin(\phi) \cos(\phi) \dot{\theta} \dot{\phi} \\ j \sin(\phi) \cos(\phi) \dot{\theta}^2 \end{bmatrix}. \quad (\text{B.8})$$

The inertia matrix \mathbf{J} is given by:

$$\mathbf{J} = \begin{bmatrix} m\mathbf{I}_3 & \mathbf{0} \\ \mathbf{0} & j\mathbf{K} \end{bmatrix}, \quad (\text{B.9})$$

with $j = \frac{ml^2}{12}$.

Define the mass matrix \mathbf{M} :

$$\mathbf{M} = \begin{bmatrix} m\mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}, \quad (\text{B.10})$$

which is trivially positive definite.

And the following matrix:

$$\mathbf{H}(\mathbf{q}) = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & j^{-1}\mathbf{K}^{-1} \end{bmatrix}. \quad (\text{B.11})$$

Finally the equations of motion are given by:

$$\mathbf{M}\ddot{\mathbf{q}} - \mathbf{H}(\mathbf{q})\mathbf{g}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{H}(\mathbf{q})\mathbf{f}_q(\mathbf{q}), \quad (\text{B.12})$$

or if $\mathbf{H}(\mathbf{q})$ is non-singular:

$$\mathbf{H}(\mathbf{q})^{-1}\mathbf{M}\ddot{\mathbf{q}} - \mathbf{g}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{f}_q(\mathbf{q}). \quad (\text{B.13})$$

If we can show that the generalized forces $\mathbf{f}_q(\mathbf{q})$ can take any value in \mathbb{R}^5 , then the system is feedback linearizable with the condition $\phi \in]0, \pi[$, because with the change of variable $\mathbf{f}_q(\mathbf{q}) = \mathbf{H}(\mathbf{q})^{-1}\mathbf{M}\mathbf{v} - \mathbf{g}(\mathbf{q}, \dot{\mathbf{q}})$ we get:

$$\ddot{\mathbf{q}} = \mathbf{v}. \quad (\text{B.14})$$

Wroldsen shows that the generalized forces can be written as:

$$\mathbf{f}_q(\mathbf{q}) = -\Phi^T(\mathbf{q})\mathbf{f}_n, \quad (\text{B.15})$$

where $\Phi(\mathbf{q})$ is a matrix containing the partial derivatives of the nodal coordinate vector $[\mathbf{n}\mathbf{0}^T \mathbf{n}\mathbf{1}^T]$ w.r.t. the generalized coordinates \mathbf{q} . This Jacobian matrix is given by:

$$\Phi(\mathbf{q}) = \begin{bmatrix} 1 & 0 & 0 & 1/2 \sin(\phi) \sin(\theta) & -1/2 \cos(\theta) \cos(\phi) \\ 0 & 1 & 0 & -1/2 \cos(\theta) \sin(\phi) & -1/2 \cos(\phi) \sin(\theta) \\ 0 & 0 & 1 & 0 & 1/2 \sin(\phi) \\ 1 & 0 & 0 & -1/2 \sin(\phi) \sin(\theta) & 1/2 \cos(\theta) \cos(\phi) \\ 0 & 1 & 0 & 1/2 \cos(\theta) \sin(\phi) & 1/2 \cos(\phi) \sin(\theta) \\ 0 & 0 & 1 & 0 & -1/2 \sin(\phi) \end{bmatrix}. \quad (\text{B.16})$$

Under the $\phi \in]0, \pi[$ constraint, $\Phi(\mathbf{q})^T$ has full rank and thus we can always find values f_n .

Form-finding

Form-finding of tensegrity structures is not an easy problem and depending on the requirements (e.g. symmetry) different solutions are available. Let us first define the precise problem of interest. In this work, we assume the connectivity of the structure to be known. This means that the connectivity matrix \mathbf{C} (Eq. 5.5) is known as well as the bar connectivity. The bar connectivity is stored in a matrix \mathbf{B} constructed as the spring connectivity matrix \mathbf{C} . For the class one tensegrities studied in this work, \mathbf{B} can trivially be rewritten (by reordering the nodes) as:

$$\mathbf{B} = \begin{bmatrix} \mathbf{I} & -\mathbf{I} \end{bmatrix} \quad (\text{B.17})$$

The problem we face is to find an equilibrium state for a structure with given matrices \mathbf{B} and \mathbf{C} , i.e. we want to know the positions \mathbf{N} of the nodes and the force density diagonal matrix $\mathbf{\Lambda}$ such that:

$$\mathbf{C}^T \mathbf{\Lambda} \mathbf{C} \mathbf{N} = \mathbf{B}^T \mathbf{\Gamma} \mathbf{B} \mathbf{N} \quad (\text{B.18})$$

$$\lambda_i > 0, \quad (\text{B.19})$$

for some diagonal matrix $\mathbf{\Gamma}$. This simply means that without external forces the forces due to the bars and and springs are balanced. As we assume the bars to have fixed lengths and to have infinite tensile strength, this results in a net acceleration of 0 of the nodes in this equilibrium configuration. If the bars do not have infinite tensile strength, then $\mathbf{\Gamma}$ depends on the Young's modulus of the bars, which is the common approach in literature.

An important question is when this equilibrium is stable and non-degenerate? To prevent degeneracy we must assure that the solution will not lie on a plane or a line. An equilibrium configuration does not guarantee that the potential energy

has a local minimum. Stability can be investigated (up to second order) using the tangent stiffness:

$$\mathbf{K} = \frac{\partial \mathbf{f}}{\partial \mathbf{N}}, \quad (\text{B.20})$$

where \mathbf{f} contains the forces acting on the nodes. If the tangent stiffness matrix \mathbf{K} is positive semidefinite, then the structure will be stable.

The problem is now to find the force densities $\mathbf{\Lambda}$ and $\mathbf{\Gamma}$ to assure that \mathbf{K} is positive semidefinite. We applied the technique from Zhang et al. [168] which starts from a given connection matrix but without knowledge of the force densities and then iteratively updates an initial estimate of the force densities. This algorithm finds structures with 12 free variables (in 3 dimensions) for a given set of force densities. We then used CMA-ES to optimize these variables to maximize the minimum bar to bar distance.

Note that in [168], struts are assumed instead of bars. Hence after form-finding, we replace the struts with bars, which does not change the equilibrium state.

Bibliography

- [1] R. M. Alexander. Principles of Animal Locomotion. Princeton University Press, Princeton NJ, 2003. ISBN 9780691126340.
- [2] Schilling N and Carrier DR. Function of the epaxial muscles in walking, trotting and galloping dogs: implications for the evolution of epaxial muscle function in tetrapods. J Exp Biol, pages 1490–1502, 2010.
- [3] Martin S. Fischer and Hartmut. Witte. Legs evolved only at the end. Phil. Trans. R. Soc. A, 365(1850):185–198, 2007.
- [4] I. Poulakakis, J. A Smith, and M. Buehler. Modeling and experiments of untethered quadrupedal running with a bounding gait: The Scout II robot. Int. J. Rob. Res., 24:239–256, april 2005. ISSN 0278-3649.
- [5] Simon Rutishauser, Alexander Sproewitz, Ludovic Righetti, and Auke Jan Ijspeert. Passive compliant quadruped robot using central pattern generators for locomotion control. In 2008 IEEE International Conference on Biomedical Robotics and Biomechatronics, 2008.
- [6] Katie Byl, Alec Shkolnik, Sam Prentice, Nick Roy, and Russ Tedrake. Reliable dynamic motions for a stiff quadruped. In Oussama Khatib, Vijay Kumar, and George Pappas, editors, Experimental Robotics, pages 319–328. Springer Berlin / Heidelberg, 2009.
- [7] M. Anthony Lewis and George A. Bekey. Gait adaptation in a quadruped robot. Autonomous Robots, 12:301–312, 2002. ISSN 0929-5593.
- [8] K. F. Leeser. Locomotion experiments on a planar quadruped robot with articulated back spine. Master’s thesis, Massachusetts Institute of Technology, 1996.
- [9] U. Culha and U. Saranli. Comparative and functional anatomy actuated spinal joint. In 2011 IEEE International Conference on Robotics and Automation (ICRA), pages 1392–1397, May 2011. doi: 10.1109/ICRA.2011.5980176.

- [10] Rolf Pfeifer and Josh Bongard. How the Body Shapes the Way We Think: A New View of Intelligence. The MIT Press, 2006.
- [11] Mohammad Hasan Haj Molla Ali Kani, Mostafa, Derafshian, Hamed Jalaly Bidgoly, and Majid Nili Ahmadabadi. Effect of flexible spine on stability of a passive quadruped robot: Experimental results. In 2011 IEEE International Conference on Robotics and Automation (ICRA), pages 2793–2797, May 2011.
- [12] Serge Gracovetsky. The Spinal Engine. Springer, cop, 1989. ISBN 9780387820309.
- [13] Billy. Spinal coord injury resource center, 2012. URL <http://www.spinalinjury.net/index.html>.
- [14] Lauren Warnecke. The stem of aplomb!cpart two: The thoracic spine, 2012. URL <http://danceadvantage.net/2011/03/09/thoracic-spine/>.
- [15] Milton Hildebrand. Motions of the running cheetah and horse. journal of the Royal Society Interface the Royal Society, 40(4):481–495, 1959.
- [16] F Iida and R Pfeifer. Sensing through body dynamics. Robotics and Autonomous Systems, 54(8):631–640, 2006. ISSN 09218890. doi: 10.1016/j.robot.2006.03.005.
- [17] Karin G.M. Gerritsen, Anton J. van den Bogert, and Benno M. Nigg. Direct dynamics simulation of the impact phase in heel-toe running. Journal of biomechanics, 28(6):661–668, 1995.
- [18] Juergen Rummel and Andre Seyfarth. Stable running with segmented legs. Int. J. Rob. Res., 27(8):919–934, August 2008. ISSN 0278-3649.
- [19] Remi Hackert, Hartmut Witte, and Martin S Fischer. Interactions between motions of the trunk and the angle of attack of the forelimbs in synchronous gaits of the pika (*ochotona rufescens*). In Adaptive Motion of Animals and Machines, pages 69–77, 2006.
- [20] Cynthia R. Lee and Claire T. Farley. Determinants of the center of mass trajectory in human walking and running. J Exp Biol, pages 2935–2944, 1998.
- [21] Farley CT, Glasheen J, and McMahon TA. Running springs: speed and animal size. J Exp Biol, (185):71–86, 1993.
- [22] J. P. He, R. Kram, and T. A. McMahon. Mechanics of running under simulated low gravity. Journal of Applied Physiology, 71(3):863–870, 1991.

- [23] S. Siegler, R. Seliktar, and W. Hyman. Simulation of human gait with the aid of a simple mechanical model. Journal of Biomechanics, 15(6):415 – 425, 1982. ISSN 0021-9290. doi: 10.1016/0021-9290(82)90078-1.
- [24] Marcus G. Pandy. Simple and complex models for studying muscle function in walking. Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences, 358(1437):1501–1509, 2003. doi: 10.1098/rstb.2003.1338.
- [25] Felix E. Zajac, Richard R. Neptune, and Steven A. Kautz. Biomechanics and muscle coordination of human walking: Part ii: Lessons from dynamical simulations and clinical implications. Gait & Posture, 17(1):1 – 17, 2003. ISSN 0966-6362. doi: 10.1016/S0966-6362(02)00069-3.
- [26] F.J. Valero-Cuevas, H. Hoffmann, M.U. Kurse, J.J. Kutch, and E.A. Theodorou. Computational models for neuromuscular function. Biomedical Engineering, IEEE Reviews in, 2:110 –135, 2009. ISSN 1937-3333. doi: 10.1109/RBME.2009.2034981.
- [27] Simon Mochon and Thomas A. McMahon. Ballistic walking. Journal of Biomechanics, 13(1):49–57, 1980. doi: 10.1016/0021-9290(80)90007-X.
- [28] R. Blickhan. The spring-mass model for running and hopping. Journal of Biomechanics, 22(11-12):1217–1227, 1989. doi: 10.1016/0021-9290(89)90224-8.
- [29] G. A. Cavagna, N. C. Heglund, and C. R. Taylor. Mechanical work in terrestrial locomotion: two basic mechanisms for minimizing energy expenditure. American Journal of Physiology - Regulatory, Integrative and Comparative Physiology, 233(5):R243–R261, 1977.
- [30] Michael H. Dickinson, Claire T. Farley, Robert J. Full, M. A. R. Koehl, Rodger Kram, and Steven Lehman. How animals move: An integrative view. Science, 288(5463):100–106, 2000. doi: 10.1126/science.288.5463.100.
- [31] Mariano Garcia, Anindya Chatterjee, Andy Ruina, and Michael Coleman. The simplest walking model: Stability, complexity, and scaling. ASME Journal of Biomechanical Engineering, 120:281–288, 1998.
- [32] A Goswami, B Thuilot, and B Espiau. A study of the passive gait of a compass-like biped robot: Symmetry and chaos. The International Journal of Robotics Research, 17(12):1282–1301, 1998.
- [33] S. H. Collins. A three-dimensional passive-dynamic walking robot with two legs and knees. Int. J. Robot. Res., 20(7):607–615, July 2001. ISSN 0278-3649. doi: 10.1177/02783640122067561.

- [34] R J Full and D E Koditschek. Templates and anchors: neuromechanical hypotheses of legged locomotion on land. Journal of Experimental Biology, 202(Pt 23):3325–3332, 1999.
- [35] James Richard Usherwood. Why not walk faster? Biology Letters, 1(3): 338–341, 2005. doi: 10.1098/rsbl.2005.0312.
- [36] Thomas A. McMahon and George C. Cheng. The mechanics of running: How does stiffness couple with speed? Journal of Biomechanics, 23(1):65–78, 1990. doi: doi:10.1016/0021-9290(90)90042-2.
- [37] Jorge G Cham, Jonathan K Karpick, and Mark R Cutkosky. Stride period adaptation of a biomimetic running hexapod. The International Journal of Robotics Research, 23(2):141–153, 2004.
- [38] B. Andrews, B. Miller, J. Schmitt, and J.E. Clark. Running over unknown rough terrain with a one-legged planar robot. Bioinspir Biomim, 6(2):026009, 2011.
- [39] John M. Schmitt. Incorporating energy variations into controlled sagittal plane locomotion dynamics. ASME Conference Proceedings, 2007(4806X): 1627–1635, 2007. doi: 10.1115/DETC2007-35740.
- [40] H. Geyer, A. Seyfarth, and R. Blickhan. Compliant leg behaviour explains basic dynamics of walking and running. P. Roy. Soc. B - Biol. Sci., 273(1603): 2861–7, November 2006. ISSN 0962-8452. doi: 10.1098/rspb.2006.3637.
- [41] R. M. Alexander. Optimization and gaits in the locomotion of vertebrates. Physiological reviews, 69(4):1199–1227, October 1989. ISSN 0031-9333.
- [42] A. E. Minetti, L. P. Ardigo, and F. Saibene. The transition between walking and running in humans: metabolic and mechanical aspects at different gradients. Acta Physiologica Scandinavica, 150(3):315–323, 1994. ISSN 1365-201X. doi: 10.1111/j.1748-1716.1994.tb09692.x.
- [43] Stacey A Combes and Robert Dudley. Turbulence-driven instabilities limit insect flight performance. Proceedings of the National Academy of Sciences of the United States of America, 106(22):9105–8, June 2009. ISSN 1091-6490. doi: 10.1073/pnas.0902186106.
- [44] H.R. Martinez and J.P. Carbajal. From walking to running a natural transition in the slip model using the hopping gait. In Robotics and Biomimetics (ROBIO), 2011 IEEE International Conference on, pages 2163 –2168, dec. 2011. doi: 10.1109/ROBIO.2011.6181612.

- [45] Harold Roberto Martinez Salazar and Juan Pablo Carbajal. Exploiting the passive dynamics of a compliant leg to develop gait transitions. Phys. Rev. E, 83(6):066707, Jun 2011. doi: 10.1103/PhysRevE.83.066707.
- [46] J. Guckenheimer and S. Johnson. Planar hybrid systems. In Hybrid Systems II, pages 202–225, London, UK, 1995. Springer-Verlag. ISBN 3-540-60472-3.
- [47] P. T. Piiroinen and Y. A. Kuznetsov. An event-driven method to simulate Filippov systems with accurate computing of sliding motions. ACM T. Math. Software, 34(3):1–24, May 2008. ISSN 00983500. doi: 10.1145/1356052.1356054.
- [48] Steven Strogatz. Nonlinear dynamics and chaos: with applications to physics, biology, chemistry, and engineering. Studies in nonlinearity. Westview Press, paperback edition, 2000. ISBN 9780738204536.
- [49] J. W. Eaton. GNU Octave Manual. Network Theory Limited, <http://www.octave.org>, 2002. ISBN 0-9541617-2-6.
- [50] Bruce Abernethy, Alastair Hanna, and Annaliese Plooy. The attentional demands of preferred and non-preferred gait patterns. Gait Posture, 15(3): 256 – 265, 2002. ISSN 0966-6362. doi: 10.1016/S0966-6362(01)00195-3.
- [51] Yuri P Ivanenko, Francesca Sylos Labini, Germana Cappellini, Velio Macellari, Joseph McIntyre, and Francesco Lacquaniti. Gait transitions in simulated reduced gravity. Journal of applied physiology (Bethesda, Md. : 1985), 110(3):781–8, March 2011. ISSN 1522-1601. doi: 10.1152/jappphysiol.00799.2010.
- [52] Li Li and Joseph Hamill. Characteristics of the vertical ground reaction force component prior to gait transition. Research quarterly for exercise and sport, 73(3):229–237, 2002.
- [53] Ben R Whittington and Darryl G Thelen. A simple mass-spring model with roller feet can induce the ground reactions observed in human walking. J Biomech Eng, 131(1):011013, 2009. ISSN 0148-0731.
- [54] Andrea d’Avella, Philippe Saltiel, and Emilio Bizzi. Combinations of muscle synergies in the construction of a natural motor behavior. Nat. Neurosci., 6: 300–308, 2003.
- [55] Yuri P Ivanenko, Germana Cappellini, Nadia Dominici, Richard E Poppele, and Francesco Lacquaniti. Coordination of locomotion with voluntary movements in humans. J. Neurosci., 25(31):7238–53, August 2005. ISSN 1529-2401. doi: 10.1523/JNEUROSCI.1327-05.2005.

- [56] G Cappellini, Y P Ivanenko, R E Poppele, and F Lacquaniti. Motor patterns in human walking and running. *J. Neurophysiol.*, 95(6):3426–37, June 2006. ISSN 0022-3077. doi: 10.1152/jn.00081.2006.
- [57] Andrea D’Avella, Laure Fernandez, Alessandro Portone, and Francesco Lacquaniti. Modulation of phasic and tonic muscle synergies with reaching direction and speed. *J. Neurophysiol.*, 100(3):1433–54, September 2008. ISSN 0022-3077. doi: 10.1152/jn.01377.2007.
- [58] Nadia Dominici, Yuri P Ivanenko, Germana Cappellini, Andrea D’Avella, Vito Mondì, Marika Cicchese, Adele Fabiano, Tiziana Silei, Ambrogio Di Paolo, Carlo Giannini, Richard E Poppele, and Francesco Lacquaniti. Locomotor primitives in newborn babies and their development. *Science*, 334(6058):997–9, November 2011. ISSN 1095-9203. doi: 10.1126/science.1210617.
- [59] Francesco Nori. Symbolic Control with Biologically Inspired Motion Primitives. PhD thesis, University of Genova, 2005.
- [60] Cristiano Alessandro and Francesco Nori. Identification of synergies by optimization of trajectory tracking tasks. In Proc. of the fourth IEEE RAS/EMBS Int. Conf. on Biomedical Robotics and Biomechatronics, pages 924–930, June 2012.
- [61] E Todorov and Z Ghahramani. Unsupervised learning of sensory-motor primitives. In Proc. 25th Int. Conf. IEEE Eng. Med. & Biol Soc., pages 1750–1753. IEEE, 2003. ISBN 0-7803-7789-3. doi: 10.1109/IEMBS.2003.1279744.
- [62] Stefan Schaal, Jan Peters, Jun Nakanishi, and Auke Ijspeert. Learning movement primitives. In Paolo Dario and Raja Chatila, editors, Robotics Research, volume 15 of Tracts in Adv. Rob., pages 561–572. Springer, 2005. ISBN 978-3-540-23214-8.
- [63] John M. Hollerbach and Tamar Flash. Dynamic interactions between limb segments during planar arm movement. *Biol. Cybern.*, 44(1):67–77, May 1982. ISSN 0340-1200. doi: 10.1007/BF00353957.
- [64] Silvia Muceli, Andreas Trø llund Boye, Andrea D’Avella, and Dario Farina. Identifying representative synergy matrices for describing muscular activation patterns during multidirectional reaching in the horizontal plane. *J. Neurophysiol.*, 103(3):1532–42, March 2010. ISSN 1522-1598. doi: 10.1152/jn.00559.2009.
- [65] Richard R Neptune, David J Clark, and Steven A Kautz. Modular control of human walking: a simulation study. *J. Biomech.*, 42(9):1282–7, June 2009. ISSN 1873-2380. doi: 10.1016/j.jbiomech.2009.03.009.

- [66] Alexander Shkolnik and Russ Tedrake. Sample-based planning with volumes in configuration space. arXiv:1109.3145v1, September 2011.
- [67] Michael L Anderson. Embodied cognition: A field guide. Artificial Intelligence, 149(1):91–130, 2003.
- [68] Rolf Pfeifer and Josh Bongard. How the body shapes the way we think: A new view of intelligence. MIT Press, 2007. ISBN 9780262162395.
- [69] Andy Clark. Being there: Putting brain, body and world together. MIT Press, 1997.
- [70] Rodney A Brooks. Elephants don’t play chess. Robotics and Autonomous Systems, 6(1-2):3–15, 1990. ISSN 09218890. doi: 10.1016/S0921-8890(05)80025-9.
- [71] Rodney A Brooks. Intelligence without representation. Artificial Intelligence, 47(1811):139–159, 1991. ISSN 1364503X. doi: 10.1098/rsta.2003.1257.
- [72] C. Paul. Morphological computation:: A basis for the analysis of morphology and control requirements. Robotics and Autonomous Systems, 54(8):619–630, 2006. ISSN 0921-8890.
- [73] D N Beal, F S Hover, M S Triantafyllou, J C Liao, and G V Lauder. Passive propulsion in vortex wakes. Journal of Fluid Mechanics, 549(-1):385, 2006. ISSN 00221120. doi: 10.1017/S0022112005007925.
- [74] Eva Kanso and Paul K Newton. Passive locomotion via normal-mode coupling in a submerged spring-mass system. Journal of Fluid Mechanics, 641:205, 2009. ISSN 00221120. doi: 10.1017/S0022112009992357.
- [75] S O Mast. Mechanics of locomotion in amoeba. Proceedings of the National Academy of Sciences of the United States of America, 9(7):258–261, 1923.
- [76] M Yanai, C M Kenyon, J P Butler, P T Macklem, and S M Kelly. Intracellular pressure is a motive force for cell motion in *Amoeba proteus*. Cell Motility and the Cytoskeleton, 33(1):22–29, 1996.
- [77] Mario De Bono and Andres Villu Maricq. Neuronal substrates of complex behaviors in *C. elegans*. Annual Review of Neuroscience, 28(28):451–501, 2005. ISSN 0147006X. doi: 10.1146/annurev.neuro.27.070203.144259.
- [78] T C Ferrée and S R Lockery. Computational rules for chemotaxis in the nematode *C. Elegans*. Journal of Computational Neuroscience, 6(3):263–277, 1999.

- [79] Francisco J Valero-Cuevas, Jae-Woong Yi, Daniel Brown, Robert V McNamara, Chandana Paul, and Hod Lipson. The tendon network of the fingers performs anatomical computation at a macroscopic scale. IEEE Transactions on Biomedical Engineering, 54(6 Pt 2):1161–1166, 2007.
- [80] H Barbeau and S Rossignol. Recovery of locomotion after chronic spinalization in the adult cat. Brain Research, 412(1):84–95, 1987.
- [81] S Grillner and P Zangger. On the central generation of locomotion in the low spinal cat. Experimental Brain Research, 34(2):241–261, 1979.
- [82] P J Whelan. Control of locomotion in the decerebrate cat. Progress in Neurobiology, 49(5):481–515, 1996.
- [83] J Tani. An interpretation of the ‘self’ from the dynamical systems perspective: A constructivist approach. Journal of Consciousness Studies, 5(5-6): 516–542, 1998.
- [84] Margaret Wilson. Six views of embodied cognition. Psychonomic Bulletin & Review, 9(4):625–636, 2002. ISSN 10699384.
- [85] Helmut Hauser, Auke J. Ijspeert, Rudolf M. Fuchslin, Rolf Pfeifer, and Wolfgang Maass. The role of feedback in morphological computation with compliant bodies. submitted to Biological Cybernetics, 2012.
- [86] H. Hauser, A.J. Ijspeert, R.M. Fuchslin, R. Pfeifer, and W. Maass. Towards a theoretical foundation for morphological computation with compliant bodies. Biol. Cybern., 105:355–370, 2011.
- [87] H Jaeger. The ‘echo state’ approach to analysing and training recurrent neural networks. Technical Report GMD report 148, German National Research Center for Information Technology, 2001.
- [88] W Maass, T Natschlag, and H Markram. Real-time computing without stable states: A new framework for neural computation based on perturbations. Neural Computation, 14(11):2531–2560, 2002.
- [89] D Verstraeten, B Schrauwen, M D’Haene, and D Stroobandt. 2007 special issue: An experimental unification of reservoir computing methods. Neural Networks, 20(3):391–403, 2007.
- [90] Ken Caluwaerts and Benjamin Schrauwen. The body as a reservoir: locomotion and sensing with linear feedback. In Rolf Pfeifer, Hidenobu Sumioka, Rudolf M. Fuchslin, Helmut Hauser, Kohei Nakajima, and Shuhei Miyashita, editors, 2nd International Conference on Morphological Computation, pages 45–47, Venice, 2011.

- [91] C Paul, J W Roberts, H Lipson, and F J Valero Cuevas. Gait production in a tensegrity based robot. ICAR 05 Proceedings 12th International Conference on Advanced Robotics 2005, pages 216–222, 2005. doi: 10.1109/ICAR.2005.1507415.
- [92] R. B. Fuller. Synergetics: Explorations in the Geometry of Thinking. Scribner, January 1975. ISBN 002541870X.
- [93] K. Snelson. Continuous tension, discontinuous compression structures, 1965.
- [94] R. Motro. Tensegrity: Structural systems for the future. Butterworth-Heinemann, 2003.
- [95] C. Sultan. Tensegrity: 60 years of art, science, and engineering. In E. van der Giessen and H. Aref, editors, Advances in Applied Mechanics, chapter 2, pages 69–145. Elsevier, 43 edition, 2009. ISBN 9780123748133. doi: 10.1016/S0065-2156(09)43002-3.
- [96] R Connelly. Tensegrity Structures: Why are they stable. In M F Thorpe and P M Duxbury, editors, Rigidity Theory and Applications, pages 47–54. Plenum Press, New York, 1999.
- [97] Robert E. Skelton and Mauricio C. de Oliveira. Tensegrity systems. Springer, 2009.
- [98] Anders Sunde Wroldsen. Modelling and control of tensegrity structures. PhD thesis, Norwegian University of Science and Technology, 2007.
- [99] C.R. Calladine. Buckminster Fuller’s “Tensegrity” structures and Clerk Maxwell’s rules for the construction of stiff frames. International Journal of Solids and Structures, 14(2):161–172, January 1978. ISSN 00207683. doi: 10.1016/0020-7683(78)90052-5.
- [100] S D Guest. The stiffness of tensegrity structures. IMA Journal of Applied Mathematics, 76(1):57–66, 2010. ISSN 02724960. doi: 10.1093/imamat/hxq065.
- [101] R Connelly and A Back. Mathematics and tensegrity. American Scientist, 86(2):142–151, 1998. ISSN 00030996. doi: 10.1511/1998.2.142.
- [102] C. Sultan, M. Corless, and R. E. Skelton. Linear dynamics of tensegrity structures. Engineering Structures, 24(6):671–685, 2002. doi: [http://dx.doi.org/10.1016/S0141-0296\(01\)00130-4](http://dx.doi.org/10.1016/S0141-0296(01)00130-4).
- [103] John A Rieffel, Francisco J Valero-Cuevas, and Hod Lipson. Morphological communication: Exploiting coupled dynamics in a complex mechanical structure to achieve locomotion. Journal of the Royal Society Interface the Royal Society, 7(45):613–621, 2010.

- [104] Thomas K Bliss. Central pattern generator control of a tensegrity based swimmer. PhD thesis, University of Virginia, 2011.
- [105] Donald E Ingber. Tensegrity: The architectural basis of cellular mechanotransduction. Annual Review of Physiology, 59(1):575–599, 1997.
- [106] D E Ingber. Tensegrity I. Cell structure and hierarchical systems biology. Journal of Cell Science, 116(7):1157–1173, 2003. ISSN 00219533. doi: 10.1242/jcs.00359.
- [107] Ning Wang, Jessica D Tytell, and Donald E Ingber. Mechanotransduction at a distance: Mechanically coupling the extracellular matrix with the nucleus. Nature Reviews Molecular Cell Biology, 10(1):75–82, 2009.
- [108] D E Ingber. Tensegrity II. How structural networks influence cellular information processing networks. Journal of Cell Science, 116(8):1397–1408, 2003. ISSN 00219533. doi: 10.1242/jcs.00360.
- [109] Auke Jan Ijspeert. Central pattern generators for locomotion control in animals and robots: a review. Neural Networks, 21(4):642–653, 2008. doi: 10.1016/j.neunet.2008.03.014.
- [110] Kiyotoshi Matsuoka. Sustained oscillations generated by mutually inhibiting neurons with adaptation. Biological Cybernetics, 52(6):367–376, 1985.
- [111] K Matsuoka. Mechanisms of frequency and pattern control in the neural rhythm generators. Biological Cybernetics, 56(5-6):345–353, 1987.
- [112] D E Rumelhart, G E Hinton, and R J Williams. Learning internal representations by error propagation. In D E Rumelhart and J L McClelland, editors, Parallel Distributed Processing, volume 1, chapter 8, pages 318–362. MIT Press, 1986. ISBN 026268053X.
- [113] Herbert Jaeger and Harald Haas. Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication. Science, 304(5667):78–80, 2004.
- [114] Kristof Vandoorne, Joni Dambre, David Verstraeten, Benjamin Schrauwen, and Peter Bienstman. Parallel reservoir computing using optical amplifiers. IEEE Transactions on Neural Networks, 22(9):1469–1481, 2011.
- [115] L Appeltant, M C Soriano, G Van Der Sande, J Danckaert, S Massar, J Dambre, B Schrauwen, C R Mirasso, and I Fischer. Information processing using a single dynamical node as complex system. Nature Communications, 2(13 September 2011):468, 2011. doi: 10.1038/ncomms1476.

- [116] J. Dambre, D. Verstraeten, B. Schrauwen, and S. Massar. Information processing capacity of dynamical systems. Scientific Reports, 2(514), 2012.
- [117] Wolfgang Maass, Prashant Joshi, and Eduardo D Sontag. Computational aspects of feedback in neural circuits. PLoS Computational Biology, 3(1):20, 2007.
- [118] Thomas Kailath, Ali H Sayed, and Babak Hassibi. Linear estimation. Number 3 in Information and System Sciences. Prentice Hall, 2000. ISBN 0130224642.
- [119] David Sussillo and L F Abbott. Generating coherent patterns of activity from chaotic neural networks. Neuron, 63(4):544–557, 2009.
- [120] Francis Wyffels, Benjamin Schrauwen, and Dirk Stroobandt. Stable output feedback in reservoir computing using ridge regression. International Conference on Artificial Neural Networks ICANN, pages 808–817, 2008. doi:10.1007/978-3-540-87536-9_83.
- [121] Ila R Fiete and H Sebastian Seung. Gradient learning in spiking neural networks by dynamic perturbation of conductances. Physical Review Letters, 97(4):5, 2006.
- [122] G. M. Hoerzer, R. Legenstein, and W. Maass. Emergence of complex computational structures from chaotic neural networks through reward-modulated Hebbian learning. Submitted, 2012.
- [123] W Schultz. Multiple reward signals in the brain. Nature Reviews Neuroscience, 1(3):199–207, 2000. ISSN 1471003X.
- [124] Robert Legenstein, Steven M Chase, Andrew B Schwartz, and Wolfgang Maass. A reward-modulated Hebbian learning rule can explain experimentally observed network reorganization in a brain control task. Journal of Neuroscience, 30(25):8400–8410, 2010.
- [125] Yonatan Loewenstein and H Sebastian Seung. Operant matching is a generic outcome of synaptic plasticity based on the covariance between reward and neural activity. Proceedings of the National Academy of Sciences of the United States of America, 103(41):15224–15229, 2006.
- [126] Yonatan Loewenstein. Robustness of learning that is based on covariance-driven synaptic plasticity. PLoS Computational Biology, 4(3):10, 2008.
- [127] D O Hebb. The organization of behavior. Wiley, 1949.
- [128] Andrew A Biewener. Animal locomotion. Oxford University Press, 2003. ISBN 9780198500223.

- [129] Nikolaus Hansen. The CMA evolution strategy: A comparing review. In J A Lozano, P Larrañaga, I Inza, and E Bengoetxea, editors, Towards a New Evolutionary Computation, volume 102 of Studies in Fuzziness and Soft Computing, chapter 4, pages 75–102. Springer, 2006. ISBN 9783540290063. doi: 10.1007/3-540-32494-1_4.
- [130] Francis Wyffels, Michiel D’Haene, Tim Waegeman, Ken Caluwaerts, Conrado Nunes, and Benjamin Schrauwen. Realization of a passive compliant robot dog. In Mamoru Mitsuishi and Masakatsu G Fujie, editors, 3rd IEEE RAS & EMBS International conference on Biomedical Robotics and Biomechanics, pages 882–886, Tokyo, 2010. IEEE. ISBN 9781424477074.
- [131] Marc Ziegler, Fumiya Iida, and Rolf Pfeifer. ”Cheap” underwater locomotion: Roles of morphological properties and behavioural diversity. In International Conference on Climbing and Walking Robots and the Support Technologies for Mobile Machines, 2006.
- [132] M. Reis, N. Maheshwari, and F. Iida. Self-organization of robot walking and hopping based on free vibration. In Rolf Pfeifer, Hidenobu Sumioka, Rudolf M. Füchslin, Helmut Hauser, Kohei Nakajima, and Shuhei Miyashit, editors, 2nd International Conference on Morphological Computation, pages 90–92, Venice, 2011.
- [133] Scott C Lenaghan, Corinne A Davis, William R Henson, Zhili Zhang, and Mingjun Zhang. High-speed microscopic imaging of flagella motility and swimming in *Giardia lamblia* trophozoites. Proceedings of the National Academy of Sciences of the United States of America, 108(34):E550–E558, 2011.
- [134] David L Ringo. Flagellar motion and fine structure of the flagellar apparatus in *Chlamydomonas*. The Journal of Cell Biology, 33(3):543–571, 1967.
- [135] Hiroyuki Terashima, Seiji Kojima, and Michio Homma. Flagellar motility in bacteria structure and function of flagellar motor. International Review of Cell and Molecular Biology, 270(08):39–85, 2008.
- [136] Kai M Thormann and Anja Paulick. Tuning the flagellar motor. Microbiology, 156(Pt 5):1275–1283, 2010.
- [137] J. Buchli and A. J. Ijspeert. Self-organized adaptive legged locomotion in a compliant quadruped robot. Autonomous Robots, 25(4):331–347, 2008.
- [138] Y. Fukuoka, H. Katabuchi, and H. Kimura. Dynamic locomotion of quadrupeds Tekken 3&4 using simple navigation. Int. J. Rob. Res., 24: 239–256, april 2005. ISSN 0278-3649.

- [139] Katie Byl and Russ Tedrake. Dynamically diverse legged locomotion for rough terrain. In IEEE International Conference on Robotics and Automation, ICRA '09., pages 1607–1608, may 2009.
- [140] S. Gracovetsky. An hypothesis for the role of the spine in human locomotion: A challenge to current thinkingal engineering. 7(3):205 – 216, 1985. ISSN 0141-5425. doi: 10.1016/0141-5425(85)90021-4.
- [141] S.A. Gracovetsky and S. Iacono. Energy transfers in the spinal engine. Journal of Biomedical Engineering, 9(2):99 – 114, 1987. ISSN 0141-5425. doi: 10.1016/0141-5425(87)90020-3.
- [142] B. M. Boszczyk, A. A. Boszczyk, and P. Reinhard. Comparative and functional anatomy of the mammalian lumbar spine. The Anatomical Record, 264(2):157–168, 2001. ISSN 1097-0185.
- [143] Rolf Pfeifer, Et Al, Rolf Pfeifer, Max Lungarella, and Fumiya Iida. Self-organization, embodiment, and biologically inspired robotics. Science, pages 1088–1093, 2007.
- [144] K. Nakajima, T. Li, H. Sumioka, M. Cianchetti, and R. Pfeifer. Information theoretic analysis on a soft robotic arm inspired by the octopus. In 2011 IEEE International Conference on Robotics and Biomimetics (Robio), pages 110–117, dec. 2011.
- [145] H. Sumioka, Y. Yoshikawa, and M. Asada. Reproducing interaction contingency toward open-ended development of social actions: Case study on joint attention. Autonomous Mental Development, IEEE Transactions on, 2(1): 40 –50, march 2010. ISSN 1943-0604. doi: 10.1109/TAMD.2010.2042167.
- [146] Yamada Yasunori, Nishikawa Satoshi, Shida Kazuya, Niiyama Ryuma, and Kuniyoshi Yasuo. Neural-body coupling for emergent locomotion: A musculoskeletal quadruped robot with spinobulbar model. In Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on, pages 1499 –1506, sept. 2011. doi: 10.1109/IROS.2011.6094752.
- [147] Max Lungarella and Olaf Sporns. Mapping information flow in sensorimotor networks. PLoS Comput Biol, 2(10):e144, 10 2006. doi: 10.1371/journal.pcbi.0020144.
- [148] Max Lungarella, Teresa Pegors, Daniel Bulwinkle, and Olaf Sporns. Methods for quantifying the informational structure of sensory and motor data. Neuroinformatics, 3:243–262, 2005. ISSN 1539-2791. 10.1385/NI:3:3:243.
- [149] N. Schmidt, M. Hoffmann, K. Nakajima, and R. Pfeifer. Bootstrapping perception using information theory: Case studies in a quadruped robot running on different grounds. Advances in Complex Systems, 2012 (acctped).

- [150] Q. Zhao, H. Sumioka, and R. Pfeifer. The effect of morphology on the spinal engine driven locomotion in a quadruped robot. In AMAM2011, pages 51–52, oct. 2011.
- [151] Qian Zhao, Hidenobu Sumioka, and Rolf Pfeifer. The effect of robot morphology on locomotion from the perspective of spinal engine in a quadruped robot. In International Conference on Morphological Computation, pages 130–132, sep. 2011.
- [152] R.Pashman and T.Kim. Normal spinal anatomy, 2012. URL <http://www.espine.com/anatomy-normal.htm>.
- [153] R. Mcn. Alexander and A. S. Jayes. Estimates of the bending moments exerted by the lumbar and abdominal muscles of some mammals. Journal of Zoology, 194(3):291–304, 1981. ISSN 1469-7998.
- [154] Arduino team. Arduino, 2012. URL www.arduino.cc.
- [155] C. Bandt and B. Pompe. Permutation entropy: a natural complexity measure for time series. Physical Review Letters, 88:174102, 2002.
- [156] J. M. Amigo, M. B. Kennel, and L. Kocarev. The permutation rate equals the metric entropy rate for ergodic information sources and ergodic dynamical systems. Physica D, 2010:77–95, 2005.
- [157] Taichi Haruna and Kohei Nakajima. Permutation complexity via duality between values and orderings. Physica D, 240:1370–1377, 2011. doi: 10.1016/j.physd.2011.05.019.
- [158] T. M. Cover and J. A. Thomas. Elements of Information Theory. John Wiley Sons, New York, 1991.
- [159] T. Schreiber. Measuring information transfer. Physical Review Letters, 85:461–464, 2000.
- [160] Matthaus Staniek and Klaus Lehnertz. Symbolic transfer entropy. Physical Review Letters, 100:158101(4), 2008.
- [161] T. Haruna and K. Nakajima. Symbolic transfer entropy rate is equal to transfer entropy rate for bivariate finite-alphabet stationary ergodic markov processes. arXiv:1112.2493v2, 2011.
- [162] T. Haruna and K. Nakajima. Permutation complexity and coupling measures in hidden markov models. arXiv:1204.1821v2, 2012.
- [163] Qi Deng, Shigang Wang, Wei Xu, Jinqiu Mo, and Qinghua Liang. Quasi passive bounding of a quadruped model with articulated spine. Mechanism and Machine Theory, 52(0):232 – 242, 2012. ISSN 0094-114X.

- [164] T Sanger. Optimal unsupervised learning in a single-layer linear feedforward neural network. Neural Networks, 2(6):459–473, 1989. ISSN 08936080. doi: 10.1016/0893-6080(89)90044-0.
- [165] E Oja. A simplified neuron model as a principal component analyzer. Journal of Mathematical Biology, 15(3):267–273, 1982.
- [166] Peter N Brown, George D Byrne, and Alan C Hindmarsh. VODE: A Variable-Coefficient ODE Solver. SIAM Journal on Scientific and Statistical Computing, 10(5):1038, 1989. ISSN 01965204. doi: 10.1137/0910062.
- [167] K Yamane and Y Nakamura. Stable penalty-based model of frictional contacts. In IEEE International Conference on Robotics and Automation, ICRA, pages 1904–1909. IEEE, 2006. ISBN 0-7803-9505-0. doi: 10.1109/ROBOT.2006.1641984.
- [168] J Zhang and M Ohsaki. Adaptive force density method for form-finding problem of tensegrity structures. International Journal of Solids and Structures, 43(18-19):5658–5673, 2006. ISSN 00207683. doi: 10.1016/j.ijsolstr.2005.10.011.