Design principles and learning schemes for applied morphological computation in robotics under consideration of aspects of material science

Authors: Harold Roberto Martinez Salazar (UZH)- Chapter: 1, 4
Qian Zhao (UZH)- Chapter: 1, 3, 9
Cristiano Alessandro(UZH)- Chapter: 1, 6, 7
Naveen Kuppuswamy(UZH)- Chapter: 1, 5, 8
Hung Vu (UZH)- Chapter: 1, 2
Rolf Pfeifer (UZH)- Chapter: 1, 2
Xiaoxiang Yu (UZH)- Chapter: 2
Fumiya Iiday (ETH)- Chapter: 2
Timothy Sandy (UZH)- Chapter: 3
Ioannis Delis (IIT)- Chapter: 6
Bastien Berret (UPS)- Chapter: 6
Stefano Panzeri (UG)- Chapter: 6
Francesco Nori (IIT)- Chapter: 6
Andrea d’Avella (SLF)- Chapter: 7
Juan Pablo Carbajal (UGENT)- Chapter: 7
# Contents

1 Introduction .................................................. 4

2 Improving energy efficiency of a hopping robot via joint stiffness variability .......................... 8
   2.1 Introduction .................................................. 8
   2.2 A segmented leg model in hopping ......................... 10
       2.2.1 Numerical model of a segmented leg ................. 10
       2.2.2 Simulation results and analysis ...................... 13
   2.3 A mechanism to improve energy efficiency ............... 18
       2.3.1 Mechanical design of L-MESTRAN ................. 18
       2.3.2 Principle of stiffness variability in L-MESTRAN .... 21
       2.3.3 Identification of the torque-deflection relationship ... 24
   2.4 Hopping experiments ....................................... 25
       2.4.1 Segmented leg hopping platform ..................... 26
       2.4.2 CoT versus fixed joint stiffness ................. 27
       2.4.3 Increasing energy efficiency during hopping ....... 29
   2.5 Discussions ................................................ 32
   2.6 Conclusions ............................................... 34

3 Development of Pneumatically Driven Quadruped Robot with Actuated Spine for Cheetah-like Behavior 37
   3.1 Introduction .................................................. 37
   3.2 Pneumatic Actuation System ............................... 38
   3.3 Prototypes and Early Testing ............................... 41
   3.4 Robot Design ................................................ 43
       3.4.1 Leg Design ............................................. 44
       3.4.2 Spine Design .......................................... 46
       3.4.3 Control Design ....................................... 47
   3.5 Results ...................................................... 47
   3.6 Future Work ................................................. 49
   3.7 Conclusion .................................................. 50
7 Synthesis and Adaptation of Effective Motor Synergies via Dynamic Response Decomposition

7.1 Introduction

7.2 Definitions and Methods

7.2.1 Solution to reaching tasks: the dynamic responses decomposition

7.2.2 Synthesis and Development of Synergies

7.3 Results

7.3.1 Point-to-point reaching tasks

7.3.2 Via-point reaching tasks

7.3.3 Reduction vs. PCA

7.4 Discussion

7.5 Conclusion and Future Work

8 Synthesising a Reduced Dimensional Oculomotor controller

8.1 Introduction

8.2 Methods

8.2.1 Simplified Linear Model of Oculomotor Dynamics

8.2.2 Spontaneous Motor Activity

8.2.3 Dimensionality Reduction by Balancing

8.2.4 Biological Motor Primitives

8.2.5 Motor-primitive based Oculomotor control

8.2.6 Minimum Dimensional Primitive based Fixation Control

8.3 Experiments and Results

8.3.1 Control of Saccading and fixation

8.3.2 Effect of Orbital Tissue Damping

8.4 Discussion

9 Spine dynamics as a computational source in spine-driven quadruped locomotion

9.1 Introduction

9.2 Experimental Setting

9.2.1 Biologically inspired compliant spine

9.2.2 Information processing with spine dynamics

9.2.3 Teaching signals

9.2.4 Experiment procedure

9.3 Results

9.3.1 Versatile behavior

9.3.2 Stability

9.4 Conclusion and Discussion
Chapter 1

Introduction

In the earlier deliverables we described the notion of morphological computation (D.3.2), and we showed how it can be used to simplify control in several scenarios, i.e. quadruped locomotion, bipedal locomotion and reaching (D.3.3). Some results were presented in simulation while others were demonstrated in the Kitty experimental platform. In this deliverable we continue with the simulation and experimental studies. In particular, we formalize design principles for exploitation of natural dynamics in the perspective of morphological computation. This is done under the following three view points: energy efficiency, robustness, and reduced dimensionality. We also demonstrate two learning mechanisms; one applied on a developmental model for human oculomotor control, and the other, based on reservoir computing, tested on Kitty robot.

In chapter two, we propose a method to improve energy efficiency of a segmented hopping leg. The method is based on appropriately regulating joint stiffness in relation to the actuating frequency. Simulation results indicate that for each frequency of actuation, a maximum energy efficiency can be achieved for an optimal joint stiffness. This is validated on a physical platform consisting of a single leg on a boom actuated by a harmonic oscillator. This study demonstrates that morphological properties can be exploited to improve the performance measure of energy efficiency.

In chapter three, we study how material properties can be exploited to facilitate locomotion. We analyze the impact of spine morphology on gait patterns using the quadrupedal robot Renny. This platform is powered by McKibben-type pneumatic actuators, which were chosen for their bio-mimetic characteristics, and their adjustable stiffness. We observe that the control of the spine stiffness increases the stride length, as compared to a fixed stiffness. This is due to an increment of the flexion and extension spinal movements allowing the legs to swing further.
In chapter four, the spring loaded inverted pendulum (SLIP) model with constant angle of attack is used as a unified framework to analyze bipedal locomotion. We redefine this model in terms of three non-dimensional variables. This reformulation generalizes the equation of motion, and allows comparisons of experimental data across human subjects. We performed experiments on a treadmill, and collected data for three loading conditions during walking and running for each subject. Results show that different morphological properties covary to compensate for the variability in body mass resulting in the same non-dimensional dynamical system. Therefore the regulation of the angle of attack seems to be a control strategy which is inherently robust against changes of mass. As a general design principle, we propose that robustness of behavior can be achieved by incorporating morphological properties within a control strategy.

The experiments in chapter five offer another viewpoint of morphological computation relevant to learning, that of reduced dimensionality. This is inspired from biological evidence, which hypothesize that neural control mechanisms implement reduced dimensional strategies to cope with the large dimensionality of the musculoskeletal system. Such strategies simplify learning and optimization of motor control. We thus hypothesize that reduced dimensionality is an inherent principle in a system exploiting its morphology.

This principle of reduced dimensionality is demonstrated in two studies. The first shows how some material properties contribute to reduced dimensionality in a simulated spring mass chain. A reduced dimensional controller, guaranteeing a given performance threshold, is synthesized from the natural dynamics. This synthesis is then repeated for variations in material properties (mass, stiffness and damping) simulating growth. The results indicate that under some conditions (low stiffness coupled with large damping) a lower dimensional control scheme can satisfy performance measures on accuracy. This indicates a general design principle on how material properties can be effectively varied to assist the control learning process.

The second study builds upon the hypothesis of muscle synergies. Neuroscientific evidence suggests that the central nervous system (CNS) implements a dimensionality reduction by restricting muscle contractions (i.e. input signals) to combinations of predefined patterns (i.e. synergies or primitives). We develop a computational quantification of the reduction obtained by pattern based control strategies in linear structures (demonstrated with a tethered mass model). Our computational framework allows characterization of task trajectories in terms of reduced dimensionality. Examples of optimal trajectories satisfying task constraints are demonstrated.

In chapter six the notion of muscle synergies is reviewed and analyzed in
Research that was carried out in the context of robotics and computational motor control highlights the importance of assessing the quality of a set of synergies at the task level. This is in opposition to the main approach employed in experimental neuroscience, that solely evaluates the capability of synergies to reconstruct a dataset of input signals (i.e. muscle activations).

The Dynamic Response Decomposition (DRD), explained in Chapter seven, clarifies this difference even further. DRD is a method to generate open-loop controllers that are defined as linear combinations of predefined actuations (synergies). Additionally, it naturally provides the agent with a strategy to autonomously synthesize such actuations. The synthesized synergies are evaluated by means of different types of errors: projection error, which measures how well the synergies approximate the desired input signals (actuations), and forward dynamics error which quantifies the task performance error. Results clearly show that the former can only be used as a heuristic measure, thus supporting the observation presented that a task based evaluation is needed to guarantee the desired performance level (Ch. 6).

In deliverable 3.3 DRD was shown to be successful in controlling a simulated kinematic chain for point-to-point reaching tasks. Here we show its application to via-point tasks. Additionally, we provide a more detailed mathematical formulation, which offers a threshold on the minimum number of synergies. The synthesized synergies are essentially the solution to a handful of representative motor tasks. Since this control scheme inherently incorporates the dynamical properties of the system, it can be regarded as a form of morphological computation.

Chapters eight and nine introduce two learning schemes that are relevant to morphological computation. Chapter eight proposes a developmental model of motion learning in the human eye. This model is based on the design principle of reduced dimensionality, and offers a general framework which can replicate previously proposed control schemes. Additionally, it suitably exploits the morphological and dynamical properties of the system. In our work, the horizontal eye movement is perturbed by a form of Spontaneous Motor Activity (SMA), and the resultant state trajectories are acquired. We use projection-based techniques to reduce the dimensionality of the state space. The resulting reduced dimensional model is utilized to compute optimal controllers for saccadic reaching tasks (high speed reaching).

In chapter nine, reservoir computing is used as a form of morphological computation. This notion is implemented by using the spine of the Kitty robot as a physical reservoir. The spine is an actuated multi-joint structure consisting of a sequence of compliant silicon blocks, and its dynamics is captured by a set of force sensors. In the previous deliverables we showed that
this spine can be used as a computational resource, where motor commands and sensor readings represent its input and output respectively. The outputs are then linearly combined to generate a desired signal. In D.3.2, we trained the linear readouts to learn a set of prescribed outputs. In this deliverable, we show a closed loop control architecture based on reservoir computing. This architecture is implemented as the state of art for the first time in Kitty. Herein the generated signals are used to actuate the spine with the purpose of producing a locomotion behavior. Results show that the body can generate multiple gait patterns only by adjusting the linear readout, even if the mechanical structure is fixed.
Chapter 2

Improving energy efficiency of a hopping robot via joint stiffness variability

2.1 Introduction

Improving energy efficiency while achieving versatile locomotion capability is a highly concerned research topic in legged robot development ([1]). There have been several robots that can achieve energy efficiency as comparable to humans/animals, however, they are constrained to a small range of working conditions. For instance, the robots only walk down a fixed slope, or run at predefined frequencies ([2–7]). Autonomous robots, which have to interact with unknown and changing environments, will have to be more versatile and should have the ability to adapt to new environmental conditions with minimum loss of energy efficiency.

A typical way of increasing energy efficiency in legged robots is to utilize elastic elements as an energy storage mechanism because they can store kinetic energy into potential energy and recycle it ([8]). In biological systems, it has been found that animal locomotion can be greatly beneficial from energy recoil of tendons (i.e. elastic elements) ([9], [10]). Compliant joints have also been widely used and have demonstrated their advantages in legged locomotion ([11], [12], [13], [14]). Curved beam robots are designed and controlled to exploit their own dynamics for hopping locomotion via elasticity of the beams, that archives energy efficiency as humans and animals ([6]). The elastic bow leg hopper can hop up to 50 cm in height, 1 m/s in speed and recover 70% of energy from one hop to the next ([7]). The remaining issue of these robots is that they have to be operated at some certain ac-
tuating frequencies for energy efficiency. Changing the actuating frequency could radically lower down their efficiency.

In the theory of harmonic oscillator system with springs and masses, it is well understood that if the sinusoidal driving force excites the system at its natural frequency, then the amplitude of the oscillation grows rapidly. Biological findings have also found that to achieve energy efficient running, leg systems are required to alter their natural frequencies, determining by their masses and stiffness, according to leg’s oscillating frequencies ([15],[16]). Natural frequency of segmented legs can be adjusted by their joint stiffness. For that purpose, variable stiffness actuators (VSAs) can provide solutions. VSAs are a new class of actuation systems which can regulate the system stiffness and therefore change the system’s natural frequency ([17],[18],[19],[20],[21],[22],[23],[24]). Some of VSAs have been used to improve energy efficiency of legged robots. A multi-directional variable stiffness leg for dynamic running has been used to tune the leg’s natural frequency for an effective gait control ([25]). Chobino, a 1D-hopping leg, equipped with a variable stiffness actuator, named MACCEPA, has demonstrated that by changing knee stiffness, the robot can achieve a much higher hopping height in comparison to the height obtained when a stiff actuator is used ([26]). Hurst et al. has designed and constructed a variable stiffness joint based on changing the tension of the pulley system which connects to the elastic elements. This mechanism is used in the MABEL robot to study the effects of leg stiffness in dynamical running ([27]).

Although many VSAs have been developed, there are a number challenges for applying them in highly dynamic robots. First, most of the current VSAs need a large amount of energy to change/maintain the stiffness. Second, they exhibit only a small range of reachable stiffness. Third, power density is needed to be improved. Forth, they demonstrate a strong coupling between stiffness and deflection. This coupling is due to the fact that the stiffness is varied together with the legs deflection. As a consequence, the affect of the stiffness on locomotion can not be extracted from that of other parameters. Therefore, influences of joint stiffness on energy efficiency has not been clarified.

From this perspective, in this paper, we investigate a mechanism to improve energy efficiency of hopping locomotion at the variations of actuating frequency by using a planar one-leg robot. The leg robot consists of two segments, an actuated joint at the hip, and a passive joint at the knee position which is equipped by a novel variable stiffness mechanism, name L-MESTRAN (Linear MEchanism for varying Stiffness via Transmission ANgle). The mechanism overcomes the challenges of existing VSAs and allows us to systematically investigate affects of joint stiffness on energy efficiency.
First, we develop a numerical model as well as a physical platform to carry out the study. To evaluate energy efficiency, we apply the dimensionless measurement as Cost of Transport (CoT) \([3]\). We varied the actuating frequencies from 2-6 Hz while changing the knee stiffness in a large range at each frequency. The simulation results have indicated that: (1) at each actuating frequency, there is an optimal knee stiffness for minimal CoT, and (2) energy efficiency of hopping locomotion can be greatly improved when the leg’s natural frequency determined by the knee stiffness and leg mass, is adjusted according to the leg’s actuating frequency. Second, we conduct various hopping experiments to validate these results. Furthermore, we demonstrate that L-MESTRAN, originated from the MESTRAN actuator \([28]\), is an effective mechanism to improve energy efficiency of hopping locomotion.

We organize the paper as follows: in section 2, we introduce a numerical model for the simulated leg. We analyze the simulation results in order to find relationships between the leg actuating frequencies and its joint stiffness for optimal CoT. In section 3, we present a development physical legged platform which consists of a two-segmented leg with L-MESTRAN at the knee joint. In section 4, we investigate the relationship in the real-world platform. We analyze the experimental data and compare them to the simulation data. In section 5, we discuss about the overall results of simulation and experiments. The last section is followed by conclusions.

2.2 A segmented leg model in hopping

In this section, we investigate the relationship between the leg actuating frequency and the knee stiffness for energy efficient hopping in the simulation. We first clarify the design parameters and state variables of the hopping leg. We then show the hopping behaviors and simulation results. Finally, the relationships between the actuating frequency, joint stiffness, and energy efficiency are presented and analyzed.

2.2.1 Numerical model of a segmented leg

We aim to develop a numerical model which is mechanically realistic and thus, the results can potentially be demonstrated by a physical platform. The leg model is developed as shown in Fig. 2.1(a,b). This model consists of two links and two revolute joints (actuated hip and passive knee). There are 12 design parameters \([l_1, m_1, l_2, m_2, I_1, a_1, I_2, a_2, \beta_0, m_0, K_s, b]\) and 8 state variables \([q_1, q_2, x_m, y_m, \dot{q}_1, \dot{q}_2, \dot{x}_m, \dot{y}_m]\). In particular, \(l_i, m_i, I_i, a_i, K, b, q_i, x_m, y_m, \dot{x}_m, \dot{y}_m\) denote for the \(i\)th link length, \(i\)th link mass, \(i\)th link moment of inertia, \(i\)th link joint variable, \(i\)th link position, \(i\)th link velocity, \(i\)th link acceleration, \(i\)th link position, \(i\)th link velocity, \(i\)th link acceleration. The first link is actuated by a linear actuator, and the second link is passive.
Figure 2.1: A hopping leg model. (a) Design parameters of a two-segmented leg model. The arrow crossing the passive complaint joint indicates that the joint stiffness is variable. (b) State variables and hopping motion of the two-segmented leg model in one hopping cycle. The direction of the horizontal arrow indicates the desired forward hopping. The gravity direction is illustrated by the vertical arrow $g$.

link mass, $i$th link inertia, length from joints to link’s center of mass, joint stiffness, joint damping, speed of $m_0$ mass, vertical speed of $m_0$ mass, respectively. In addition, $g$, $F_x$, $F_y$, $\tau_m$, and $\tau_k$ are the gravity acceleration, horizontal ground reaction force, vertical ground reaction force, hip actuated torque, and knee spring torque, respectively.

All state variables and hopping behaviors during one cycle are shown Fig. 2.1. One cycle of hopping locomotion consists of two phases, stance and flight phases (i.e. the foot touches and not touches the ground). The cycle starts from one apex height to the next apex height. The moment when the foot touches the ground is defined as the touch-down (TD) time point. The moment when the leg leaves the ground for the flight phase is defined as take-off (TO) time point. When the top mass, $m_0$ reaches the apex point, the height of foot with respect to the ground is defined as the hopping height $h_{\text{apex}}$. The equations of motion of the model in a generalized coordinate are as follows:

$$M(\vec{u}) \ddot{\vec{u}} + B(\vec{u}, \dot{\vec{u}}) \dot{\vec{u}} + C(\vec{u}) = \vec{\tau}, \quad (2.1)$$

where $\vec{u} = [q_1, q_2, x_m, y_m]^T$, and $\dot{\vec{u}}$ and $\ddot{\vec{u}}$ are first and second order derivative of $\vec{u}$. $M(\vec{u})$ is a $4 \times 4$ matrix of the mass and inertia, $B(\vec{u}, \dot{\vec{u}})$ is a $4 \times 4$ Coriolis/Centripetal matrix while $C(\vec{u})$ is a $4 \times 1$ gravity vector. $\vec{\tau}$ denotes a $4 \times 1$ vector of the external generalized forces, which are formulated as

$$\vec{\tau} = \vec{\tau}_m + \vec{\tau}_G + \vec{\tau}_K, \quad (2.2)$$

where $\vec{\tau}_m$, $\vec{\tau}_G$, and $\vec{\tau}_K$ are $4 \times 1$ vectors and denote the external loads such as motor torques, spring torque, and ground reaction forces, respectively.
In order to simulate the ground reaction forces, a realistic nonlinear 
spring-damper model based on (29) is applied. The horizontal ground force, 
$F_x$, is modeled as the Coulomb's friction in Eq. 2.3 and the vertical ground 
force, $F_y$, is produced by a nonlinear spring-damper model in Eq. 2.4.

\[
F_x = \begin{cases} 
F_{\text{slide}} = \mu_{sl} F_y \frac{\dot{x}_g}{|\dot{x}_g|} & \text{if } F_x > F_{\text{stiction}} \\
F_{\text{stiction}} = \mu_{st} F_y \frac{\dot{x}_g}{|\dot{x}_g|} & \text{otherwise}
\end{cases}
\]  
(2.3)

\[F_y = a |y_g|^3 (1 - b \dot{y}_g) \]  
(2.4)

where $x_g$ and $y_g$ are the coordinates of the foot with respect to the ground 
while $\dot{x}_g$ and $\dot{y}_g$ are the velocities of $x_g$ and $y_g$. $\mu_{sl}$ and $\mu_{st}$ are the kinetic 
and stiction friction coefficients. $a$ and $b$ are two empirical parameters.

Since we are interested in passive behaviors of the hopping leg in order 
to study its natural dynamics, we apply a very simple position control to 
actuate $L_1$ to track a sinusoidal position input as:

\[q_1(t) = A \sin(2\pi f t) + q_1(0), \]  
(2.5)

where $q_1(t)$ denotes the desired angles of $L_1$ at time $t$, $A$ is the amplitude 
of the sinusoidal signal and $f$ denotes the actuating frequency. In order to 
extract the influences of the join stiffness without any interference of other 
parameters, we fixed $A$ and $q_1(0)$ while varying $f$ and $K_\theta$. The initial 
conditions of the system are $q_1(0)$, $q_2(0)$, $x_m(0)$, $y_m(0)$, $\dot{q}_1(0)$, $\dot{q}_2(0)$, $\dot{x}_m(0)$ and 
$\dot{y}_m(0)$. In order to reach desired positions, PID control method is imple-
mented.

Mechanical specifications of the simulated leg were obtained from a real 
leg system (Tab. 2.1), which were then fabricated and experimented in the 
next section. We employ the Symechanics Toolbox in Matlab(Mathwork 
Inc.) to implement simulation. The actuating frequencies, $f$, are ranged 
from $2 - 6$ (Hz) with an interval of $0.25$ Hz. The stiffness of the knee joint 
is varied from $0.7-4.7$ (Nm/rad) with an interval of $0.2$ Nm/rad. For each 
combination of one actuating frequency and one joint stiffness, we simulate 
the hopping leg in a duration of 5 seconds with the Runge-Kutta solver, 
fixed-step size of $0.0002$ sec. In order to avoid undamped oscillations of the 
leg during flight, we configure the knee joint as a preloaded asymmetrical 
joint, i.e. the knee joint can only be deflected to one side. In addition, 
the preloaded torque at the joint can maintain the leg bending angle ($\beta_0$) 
unchanged at stand-still.

To evaluate and compare energy efficiency in the simulation and later in 
the real world experiments, we use the CoT as a criterion which is defined
Table 2.1: Parameters of the experiment platform used in the simulation

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>height of robot</td>
<td>0.25m</td>
</tr>
<tr>
<td>length L1</td>
<td>0.115m</td>
</tr>
<tr>
<td>length L2</td>
<td>0.155m</td>
</tr>
<tr>
<td>mass $m_0$</td>
<td>0.6kg</td>
</tr>
<tr>
<td>mass $m_1$</td>
<td>0.32kg</td>
</tr>
<tr>
<td>mass $m_2$</td>
<td>0.80kg</td>
</tr>
<tr>
<td>length $a_1$</td>
<td>0.06m</td>
</tr>
<tr>
<td>length $a_2$</td>
<td>0.1m</td>
</tr>
<tr>
<td>angle $\beta_0$</td>
<td>135°</td>
</tr>
</tbody>
</table>

as follow:

$$CoT = \frac{P}{M_{\text{total}}gv},$$

(2.6)

where $M_{\text{total}}$ is the total mass of the robot, $g$ is the gravitational acceleration, $v$ is the traveling speed, and $P$ denotes the external power input which is obtained by the product of the simulated hip torque, the supply voltage and the current constant of the DC motor, used in the physical platform.

2.2.2 Simulation results and analysis

The numerical model has been constructed to serve as an experimental platform. A number of hopping simulation were carried out. In this section, we first show the results of the simulation which are the $CoT$ versus the knee stiffness in the different actuating frequencies. Later, we analyze the variations of the $CoT$ in the relationships with the speed and the power consumption.

Hopping simulation results

The hopping behavior of the leg model at the actuating frequency of 5 Hz in 10 hopping cycles is shown in Fig. 2.2. The state variables of the hip joint, the knee joint together with the actuated torque at the hip joint, and the ground reaction forces are plotted in one cycle scale (0.2 sec) starting from one TO to the next TO. Note that the $x_m(t)$ plot only shows the relative distance from the TO position to the current position of the hip joint. A
vertical line, shown in all sub-plots, indicates the switching points between the flight and stance phases.

Although there are slight discrepancies of data in cycles in 10 cycles (Fig. 2.2), they are highly repeatable. The trajectories, shown by $q_1$, $q_2$, $\dot{q}_2$, and $\tau_k$, indicate the asymmetrical and preloaded properties of the knee joint as follows. (1) Link $L_2$ is only deflected to one side and during the flight phase, (2) $L_2$ is held un-oscillated during the flight, and (3) the externally exerted torque $\tau_k$ is always greater than zero and remains high in the flight phase. It can be seen that the hip actuation gains the energy recoiling from the compliance of the knee joint because $\dot{q}_2$ always co-varies with $\dot{q}_1$. The distance coverage $x_m$ of the leg rises up rapidly to the stable and repeatable level after one cycle. This is also indicated by the horizontal speed $\dot{x}_m$. The maximum horizontal speed is achieved in the stance phase, shortly before the TO. The vertical excursion of the top mass $y_m$ repeats the same trajectories in all cycles. The ground reaction forces have three distinctive phases: (1) flight phase: there is no force, (2) loading phase: at the first period of the stance phase, the horizontal force $F_x$ has negative sign, and the vertical force reaches maximum, (3) releasing phase: $F_x$ negates its direction and reaches maximum to produce forward thrust force while $F_y$ decreases to zero. The analysis shows that the robot behaviors can be considered as stable and the results can be used for further analysis.

The $CoT$, the joint stiffness and the actuating frequency show some interesting relationships as the following as shown in Fig. 2.3. At each frequency of actuation, $f$, there always exists an optimal joint stiffness of $K_\theta$ ($K_{\text{opt}\theta}$) for minimum $CoT$. Increasing $K_\theta$ induces higher $f$ ($f_{\text{opt}}$) for energy efficiency. At one frequency, $CoT$ decreases faster when $K_\theta$ increases from lower values to $K_{\text{opt}\theta}$ in contrast to the $CoT$ in the region where $K_\theta$ decreases from high values to $K_{\text{opt}\theta}$. Furthermore, different actuating frequencies result in different changing rate of $CoT$ as high actuating frequencies result in low variation of $CoT$ and vice versa. These results suggest that hopping locomotion at low frequencies (e.g. 2, 3 Hz) requires a more careful adjustment of $K_\theta$ for low $CoT$. In general, increasing actuating frequency can improve energy efficiency.

In order to have understanding of how the intrinsic leg dynamics (i.e. natural frequency) influences on energy efficiency as it has been found in the simulation, we derive the equation of motion of the leg at stance phase to find the relationship between the natural frequency and the optimal frequency in the simulation. The result can be used to optimize the leg structure as well as advice the leg controller for exploiting the leg dynamics.

To estimate the leg’s natural frequencies, we first linearize the equation of motion at the stance phase by the following assumptions. (1) At equilibrium
Figure 2.2: State variables, external forces, ground reaction forces and the hopping behavior of the leg model at the actuating frequency of 5 Hz.
Figure 2.3: Simulation of the hopping leg at different actuating frequencies from 2-6Hz with step size of 0.25 Hz. At each frequency, the joint stiffness is varied from 0.7-4.7(Nm/rad) with an interval of 0.2 Nm/rad. The CoT values are presented by the color bar on the right.

point, the generalized coordinate \((q_1, q_2)\) is slightly changed, therefore the matrices, \(M, B\), and \(C\) of the Eq. 2.1 can be evaluated at the equilibrium configuration: \(q_1 = q_{1e}, q_2 = q_{2e}\) where \(q_{ie}\) is the equilibrium angle of the segment \(i\) th. (2) Around the equilibrium point, the Centripetal and Coriolis effects are small and negligible, thus the first order time derivative can be omitted. Note that in the stance phase, the number of state variables in the equation of motion can be simplified to 4: \([q_1, \dot{q}_1, q_2, \dot{q}_2]\) as necessary, instead of 8 as mentioned in section 2.1. The equation of motion of the stance dynamics around the equilibrium point is:

\[
M(q_e)\ddot{q} + C(q') = \tau, \tag{2.7}
\]

where \(\tau = K(q_e - q)\), a 2 \times 1 joint torque vectors, and \(K\) is the joint stiffness matrix and \(q' = q - q_e\) is a new coordinate which is referred to the defined generalized coordinate. Rearranging the Eq. 2.7, we obtain a homogenous form of the equation of motion around the equilibrium point.

\[
M(q_e)\ddot{q} + (K + J(q_e))(q) = Kq_e + J(q_e)q_e \tag{2.8}
\]

where \(J(q_e)\) is the Jacobian matrix of the gravity vector \(C\). The details of \(M(q_e), K\) and \(J(q_e)\) are presented in Apendix A. By definition, the natural frequencies, \(\omega^2\) of the system Eq. 2.7 around the equilibrium point are the Eigen values of the matrix: \(M^{-1}(K + J(q_e))\). Thus, we obtain the Eigen
frequencies of the system: \( f = \frac{1}{2\pi} \omega \), which is the function of the joint stiffness \( K_\theta \) as shown in Fig. 2.3.

It is shown that all optimal frequencies lie closely around the Eigen frequencies. Thus, maximum energy efficiency can be achieved when the leg’s actuating are closely regulated to its natural frequencies. Since natural frequency depends on the leg mechanic structure and joint stiffness, it suggests that, on one hand, given a desired range of actuating frequencies, one can design the leg structure such that its natural frequencies is close to the actuating frequency. On the other hand, given a known leg structure, the joint stiffness can be estimated according to the actuating frequency to exploit the joint compliance. We are aware of that there are several mismatchings between the theoretical model and the simulation model. The reasons are: (1) the dynamics of legged hopping is so-called ”hybrid dynamics” that consists of the flight and stance phases, but the model is constructed for the stance phase only, (2) the Eigen frequencies are approximately estimated. However, the results are promising to be validated in a physical platform.

**Analysis of CoT, speed and power consumption**

To further understand the variations of CoT versus joint stiffness in different actuating frequencies, we analyze the speed, \( \dot{x}_m \), and the power consumption, \( P_m \) during hopping at four different actuating frequencies, i.e. 2, 3, 5 and 6 Hz because CoT is a product of power consumption and speed, as shown in Eq. 2.6. Note that we scale the speed by \( mg \) to have a better visual comparison as shown in Fig. 2.4.

The results show that the minimal CoT is always achieved at the maximum speed whereas it is not true for \( P_m \). At one frequency, regardless of the knee stiffness, the power consumption does not change as much as the speed changes. It suggests that in order to increase energy efficiency, it is more effective to optimize speed rather than the power consumption. At the higher actuating frequency (e.g. 5, 6 Hz), the influence of the joint stiffness on the CoT is less than that of the joint stiffness at lower actuating frequencies. At 2 and 3 Hz, a small change of the joint stiffness apart from the optimal value can drastically increase the CoT. Nevertheless, at 5 and 6 Hz, the speed is relatively stable against the knee stiffness variation. This can be explained by the contribution of the input power and the elastic energy recovered from the joint compliance. At low actuating frequencies (i.e. 2, 3 Hz), the power input is low due to low rotational speed. This amount of power may not be enough to lift the leg off without the contribution of the compliant joint. In principle, when the joint stiffness is not at its optimal range, the leg is not capable of exploiting the energy saving at the compliance joint. The oscillation
of the compliant joint even further cancel out a part of the input energy. The speed drops down to near zero. However, when the joint stiffness is regulated to adjust the leg’s natural frequency to its actuating frequency, the speed, shown in Fig. 2.4 increases quickly. The large variation of the speed leads to a radical changes of $CoT$ with respect to the joint stiffness regulation at low actuating frequencies. Nevertheless, at high actuating frequencies (i.e. $5,6 \, Hz$), the input power is increased. The influence of the joint stiffness on the speed is decreased. In particular with the same variation of the joint stiffness, the speeds at the higher frequencies does not change as drastically as their change in the lower frequencies. Moreover, the power consumption is not greatly affected by the joint stiffness. Therefore, the $CoT$ is more stable against the change of the joint stiffness. In general, increasing actuating frequency can lead to larger working range of joint stiffness which can maintain low $CoT$.

We summarize the simulation results as follows. First, at each actuating frequency from 2-6 (Hz), there always exists a joint stiffness for optimal $CoT$. Second, energy efficiency of the hopping leg can be significantly increased by adjusting the natural frequency of the system to the actuating frequencies. Third, the $CoT$ is more influenced by the speed rather than by the power consumption. Forth, the joint stiffness has less influence on the $CoT$ when the actuating frequency increases. And fifth, increasing actuating frequency can lead to energy efficiency. In order to test these results, we have designed, fabricated a physical platform which allows us to carry out various experiments.

2.3 A mechanism to improve energy efficiency

In order to investigate the relationship between joint stiffness and the actuating frequency for energy efficiency in reality, we develop a mechanism which can alter the joint stiffness and be integrated into a physical leg platform. In this section, we first present a newly developed leg platform which equipes a VSM, L-MESTRAN, at the knee joint. We then validate the capability of the mechanism by a number of experiments.

2.3.1 Mechanical design of L-MESTRAN

A newly developed VSM, L-MESTRAN, was fully integrated into a hopping robot platform as shown in Fig. 2.3(a,b). The hopping robot was designed to exhibit locomotion in a planar environment, which consists of an actuated and knee joints along with Link 1 and 2. The actuated joint is further
Figure 2.4: $CoT$, power consumption and speed of the simulated hopping robot at 2Hz (a), 3Hz (b), 5Hz(c),6Hz(d). For visualization purpose, the speed is scaled up by a factor of $mg$ where $m$ is the total mass of the robot and $g$ is the gravitational constant.
Figure 2.5: Working principle of the novel stiffness mechanism, L-MESTRAN. (a) Resting state of the mechanism with no preload from the spring. (b) Resting state of the mechanism with preload from the spring. (c) The mechanism under external load. $y_0$ is the natural length of the spring. $y_1$ is the length of the spring when the slope angle changes to $\theta$ under no load. $y_2$ is the length of the spring under external load $\tau \phi$. $y_3$ is the length of the spring under external load at lower stiffness. $K_s$ is the spring constant.
connected to a supporting boom as shown in the next section.

L-MESTRAN connects Link 1 and 2 via a one-dimensional rotary joint (Fig. 2.5(a)), and the joint has a unique mechanism in which the stiffness can be actively adjusted by the “stiffness motor”. Given a signal from the robot controller, the stiffness motor is capable of varying the “slope gear” via a worm transmission. The angle of slope gear $\theta$ in Fig. 2.5(b) is an important parameter that determines the stiffness of the knee, as a lower angle of $\theta$ results in an increased stiffness, and vice versa. In particular, the working principle of the mechanism is shown in Fig. 2.5(c,d,e). In the rest state, there is no external load applied on Link 2. Due to geometrical constraint, the spring compression $y_0 - y_1$ is increased with the increase of the slope angles which results in the higher preloaded torque, $\tau_\theta$. During operation, when load exerts on Link 2 and the joint rotates counter-clockwise by an angle $\varphi$, the rack gear is pushed against the slope gear which causes the compression of the spring $K_s$ by $y_0 - y_2$. In contrast, when the slope gear has a larger angle $\theta'$ (as in Fig. 2.5(e)), the amount of spring compression, $y_0 - y_3$, is smaller, which eventually causes reducing joint stiffness with respect to the rotation angle $\varphi$ of Link 2.

This variable stiffness actuator has unique properties as compared to the other actuators proposed in the past ([17]). First, it has a very large range of stiffness variability as the stiffness can be theoretically set from zero to infinity. For example, the stiffness of the joint can be zero (i.e. completely passive) when the slope gear angle $\theta$ is $\frac{1}{2}\pi$ (rad), whereas, when $\theta$ is zero, it becomes a rigid joint. Second, the mechanism requires no energy to maintain joint stiffness, which is often not the case with other actuators ([30],[31]). Mainly due to the worm gear between the stiffness motor and the slope gear, the joint operation does not influence the control of stiffness motor. Third, another unique characteristics of this actuator lies in the fact that the joint stiffness can be decoupled from the external load unlike most of the conventional variable stiffness actuators ([32],[22],[33],[18]). As explained in the next subsection, the joint compliance is linear. In order to formalize the stiffness variability with respect to the slope angle, we derive static force analysis of the mechanism in the following section.

2.3.2 Principle of stiffness variability in L-MESTRAN

The force analysis is carried out when the mechanism is at the loading state as shown in Fig. 2.6. The external torque $\tau_\varphi$ rotates the pinion gear by an angle $\varphi$ and pushes the rack gear by a force $F_\varphi$. This force is divided in two directions which are along the x axis, $F_1$ and the y axis, $F_2$. The friction coefficient, denoted as $\mu$, at all contacting surfaces is assumed to be constant
under load. As the spring is compressed by an amount of \( y \), the force balance formula along axes are:

\[
\sum F_y = 0 \Leftrightarrow F_2 = K_s y + \mu F_1, \quad (2.9)
\]

\[
\sum F_x = 0 \Leftrightarrow F_\varphi = F_1 + \mu F_2. \quad (2.10)
\]

\( F_1 \) and \( F_1 \) are coupled as:

\[
F_2 = F_1 \tan \theta. \quad (2.11)
\]

Solving the equations 2.9, 2.10 and 2.11 we obtain:

\[
\begin{align*}
F_1 &= \frac{K_s y \tan \theta}{\tan \theta - \mu}, \\
F_2 &= K y \tan \theta + \frac{\mu K_s y \tan \theta}{\tan \theta - \mu}, \\
F_\varphi &= \frac{K_s y (\mu \tan \theta + 1)}{\tan \theta^2 - \mu \tan \theta}.
\end{align*} \quad (2.12)
\]

The torque applied on the joint is calculated from the force acting between the rack gear and the pinion gear, \( F_\varphi \)

\[
\tau_\theta = F_\varphi r_G = \frac{K_s r_G y (\mu \tan \theta + 1)}{\tan \theta^2 - \mu \tan \theta}. \quad (2.13)
\]

In case of no external torque and the spring pre-compressed of \( y_0 - y_1 \), shown in Fig. 2.5, the preloaded torque applied on the joint is:

\[
\tau_\theta = \frac{K_s r_G (y_0 - y_1)(\mu \tan \theta + 1)}{\tan \theta^2 - \mu \tan \theta}. \quad (2.14)
\]
Figure 2.7: The range of joint stiffness versus the slope angles in various spring constants from 3 N/m up to infinite.

The relationship of Link angles, the rack gear, \(x\) and the compression of the spring, \(y\) is as:

\[
x = y \tan \theta = r_G \varphi.
\] (2.15)

Therefore, the relationship between the external torque \(\tau_\varphi\) and the angular deflection \(\varphi\) is:

\[
\tau_\varphi = \frac{K_s r_G^2 (\mu \tan \theta + 1)}{\tan \theta^2 - \mu \tan \theta} \varphi + \tau_\theta.
\] (2.16)

The rotational joint stiffness at the slope angle of \(\theta\) is found by differentiating the external torque with respect to the angular deflection as follows:

\[
K_\theta = \frac{\partial \tau_\varphi}{\partial \varphi} = \frac{K_s r_G^2 (\mu \tan \theta + 1)}{\tan \theta^2 - \mu \tan \theta}.
\] (2.17)

The slope angle, \(\theta\), the pinion gear radius, \(r_G\), the friction coefficient, \(\mu\) and the spring constant \(K_s\) are unchanged and are design parameters. Therefore, the joint stiffness is a constant at a given slope angle, \(\theta\).

The variation of the joint stiffness \(K_\theta\) versus the slope angle, \(\theta\), at various spring constants \(K_s\) is shown in Fig. 2.7. As the slope angle decreases from \(\pi/2\) to 0.2 (rad), the joint stiffness increases from zero to infinity regardless of the spring constants. However, the spring constants determine the progressive characteristics of the joint stiffness curve with respect to the change of the slope angle. Higher spring constants result in more consistent changes of the joint stiffness over the range of the slope angles. Lower spring constants can introduce strong “turning points” where the joint stiffness is largely changed. In order to validate the capability of L-MESTRAN, we have fabricated the leg design as shown in Fig. 2.4, and conducted various experiments as presented in the next sub-section.
Table 2.2: Experiment devices and specifications

<table>
<thead>
<tr>
<th>Items</th>
<th>Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Torque sensor</td>
<td>FUTEK MODEL TFF325/ Torque capacity: 12Nm</td>
</tr>
<tr>
<td>Joint motor</td>
<td>Faulhaber 324212CR, 24.7 W/Gear reducer 43:1</td>
</tr>
<tr>
<td>Stiffness motor</td>
<td>Maxon RE13, 1.5 W/Gear reducer 131:1</td>
</tr>
<tr>
<td>Spring constant</td>
<td>1.6 N/mm</td>
</tr>
<tr>
<td>Controller</td>
<td>Maxon EPOS2 and Atmega328P</td>
</tr>
</tbody>
</table>

2.3.3 Identification of the torque-deflection relationship

We investigate the variability of the joint stiffness with respect to the change of the stiffness regulator (i.e. the slope angle, $\theta$), in a number of the experiments. The relationship of the joint torque versus the deflection angle is analyzed with theoretical formulation. The experiments are setup as shown in Fig. 2.8.

Link 1 was mounted on the joint motor shaft in the horizontal direction to avoid effects of gravity on moving parts. The joint motor was vertically fixed on Lab frame. The end of Link 2 was fixed by the torque sensor such that the torque on Link 2 could be transferred to the torque sensor. The torque sensor was also fixed on Lab frame. The stiffness motor, which was controlled by an Atmega328 controller, regulated the slope angle for changing the joint stiffness. The joint motor is controlled by a Maxon EPOS2 controller. The experiments were carried out as follows.

The range of the slope angle to be investigated was from 0.34 – 1.36 rad.
Exerted torque \([\text{Nm}]\)

Angular deflection \([\text{rad}]\)

Figure 2.9: Theoretical (dashed line) and experimental data (bold dots) of external torques versus angular deflections at the knee joint. The slope angle varies from \(\pi/2 - \pi/10\) to \(\pi/9\) with the interval of \(\pi/18\). The plotted results are the mean values of exerted torques in each transmission angle and the error bars show deviation of exerted torques.

with a discretized step of 0.17. At each slope angle step, the joint motor varied Link 1 from 0 – 0.68 with a step size of 0.017 three times. The torque and the angular deflection were measured simultaneously by the EPOS2. The experimental relationship of the torque and angular deflection at the joint is shown in Fig. 2.9 together with theoretical results which are obtained from Eq. 2.17. The results show that the variability of the joint stiffness when the slope angle changes as a lower slope angle results in higher joint stiffness and vice versa. Furthermore, when the slope angle increases, the rate of change of the joint stiffness also increases. Lastly, the theoretical formulation of the joint stiffness, as mentioned in Fig. 2.17, can predict well the experimental results.

As we have shown in Fig. 2.3, the simulation results exhibit an interesting relationship between the joint stiffness and \(CoT\) at various actuating frequencies. It suggests that in order to improve the energy efficiency of a hopping robot, the joint stiffness should be optimized according to the actuating frequencies. Having such VSM on the leg platform, we conducted experiments to verify the findings from the simulation in the real-world platform. We present the experiment setup and results in the next section.

2.4 Hopping experiments

The goal of these experiments is to investigate the relationship between the actuating frequencies and the joint stiffness for energy efficiency. The results
A complete leg platform supported by a boom has been constructed as shown in Fig. 2.10. All specifications of the setup are presented in Tab. 2.3. The boom support is fixed on the ground. The boom with the length of 1.07 m is mounted to the actuated joint on the top of the leg platform. As mentioned in section 3, the leg is composed of two links and two revolute joints. The L-MESTRAN is integrated at the knee joint. Due to the design features, mentioned in section 3, a preloaded torque applied by the spring precompression at the knee joint to keep the segmented angle $\beta = \beta_0$ when the robot stands still. A DC motor is mounted on the boom and actuates the leg via a gear transmission with a ratio of 2:1 from the actuated joint. The shaft of the actuated joint is fixed on the boom. The foot is covered by rubber while the ground is made of wooden plates. The power consumption is measured by an current sensor ACS712 from Sparkfun Inc. There is a touch sensor, 0.2”-diameter circle from InterLink Electronics, underneath the foot to detect the flight and stance phases.

To control the robot, we apply the same controller, which was used to
simulate the leg in Matlab, for a comparable setup. We applied 5 actuating frequencies in the experiments from $2 - 6\, Hz$ with an interval of $1\, Hz$. To change the joint stiffness, we regulate the slope angles from $26 - 46^\circ$ with a step of $2^\circ$. The range of joint stiffness in the real robot covers the joint stiffness range (i.e. $0.7-4.5\, Nm/rad$) similarly as used in the simulation. The leg was commanded to track the sinusoidal trajectories at all actuating frequencies as:

$$q_1(t) = A\sin(2\pi ft) + q_{10},$$

where $A$: Amplitude of oscillation, $f$: actuating frequency, $t$: time, and $q_{10}$: initial angle of Link 1. The real values of $A$ and $q_{10}$ are heuristically found such that the leg could be able to hop forward at all actuating frequencies.

In order to evaluate energy efficiency, we use the same dimensionless quantity as used in the simulation, $CoT$ (Eq. 2.6). The experiment was designed in the similar fashion as it was implemented in the simulation. In addition, the power and speed data were only collected at the steady state of hopping, shown in Fig. 2.12.

### 2.4.2 $CoT$ versus fixed joint stiffness

We have carried out various hopping experiments on the leg platform. The leg has demonstrated stable hopping locomotion at 5 actuating frequencies and wide range of stiffness. We show the hopping behavior of the leg at the actuating frequency of $5\, Hz$ in Fig. 2.11 in two cycles. In addition, Link trajectories, the speeds, and the power consumptions for ten hopping cycles are presented in the Fig. 2.12. The flexible pressure sensor at the foot is used to separate the flight and stance phases.

The hopping series show that the robot postures are repeated after one
Figure 2.11: Time series of two hopping cycles at the hip oscillating frequency of 5Hz.

Figure 2.12: Time series of Link trajectories, speed and power consumption during 10 hopping cycles at the stride frequency of 5Hz
cycle. Link 1’s trajectories ($q_1$) during hopping cycles are repeatable, though there are slight discrepancies. The deflection of the knee joint indicates the following: (1) the joint deflection is asymmetrical, (2) during the flight phase, there is a small oscillation (smaller than $0.15 \text{ rad}$ or $8^\circ$) due to the preloaded torque, (3) the compression starts after the TD, reaches full compression in the middle of the stance and releases when the leg retracts before the flight phase. This indicates the synchronization between the knee joint deflection and the hip joint actuation in the way that the leg actuation is beneficial from the joint compliance. In 10 hopping cycles, the averaged speed of the leg ($\dot{x}_m$) is about $1.4 \text{ m/sec}$. The power consumption is repetitive between hopping cycles.

The experimental results have indicated that at all actuating frequencies (i.e., 2-6 (Hz)), the variation of the CoT versus joint stiffness in the experiments (Fig. 2.13) have shown the similarity to the simulation results (Fig. 2.3). In particular, at a given actuating frequency, there is an optimal joint stiffness for an optimal/minimal CoT, which has been shown in the simulation results. For example, at 2Hz, the CoT is optimal at a joint stiffness of 0.8Nm/rad. An increased or decreased joint stiffness apart from the optimal one results in the increased CoT. However, the higher joint stiffness always lead to more stable CoT at all actuating frequencies. At 2Hz, a small change of knee stiffness can result in a large variation of the CoT. In contrast, at 6Hz, the change of the joint stiffness does not lead to a large variation of the CoT. As mentioned in section 2, hopping at low frequency requires a higher care of joint stiffness compared to hopping at higher. Furthermore, the working range of the joint stiffness are shifted to the higher values as the actuating frequencies are increased. Lastly, increased actuating frequencies promotes lower CoT or higher energy efficiency. These experimental results have validated the findings in the simulation results that we have summarized in the end of section 2. These results indicate that to achieve energy efficiency, the robot is not constrained to some certain actuating frequencies, instead appropriately regulating the joint stiffness can always improve energy efficiency at any actuating frequency. Moreover, during operation, L-MESTRAN can be used to regulate the joint stiffness to increase energy efficiency as we will demonstrate in the following sub-section.

### 2.4.3 Increasing energy efficiency during hopping

The results of the fixed joint stiffness experiments is demonstrated by dynamically changing joint stiffness to improve energy efficiency during hopping at the actuating frequency of 4 Hz. First case, the robot starts hopping at a low stiffness and then the joint stiffness is increased to the optimally found value
Figure 2.13: CoT of the single-leg hopping in different oscillating frequencies (2-6Hz) with respect to slope angle and knee stiffness (Nm/rad). The error bars present the standard deviation. The minimum CoTs are pointed by the arrow attached with corresponding actuating frequencies.

(1.5Nm/rad—shown in Fig 2.13) as shown in Fig. 2.14a. Second, the joint stiffness is set high when the robot starts, it is later decreased to the optimal one as shown in Fig. 2.14b. At the time when the joint stiffness changes, the speed, power consumption and CoT are varied in different ways. In the first case, as the joint stiffness is started at lower value than the optimal one, the CoT is high (12), and the speed is almost zero (0.1 m/sec). However, it still consumes relatively high power. During the transition period at which the joint stiffness is commanded to increased, the CoT starts decreasing, the speed and power consumption are rise slowly. After 6 seconds, the robot achieves more stable behavior that the speed and power consumption are stabilized. The CoT is decreased radically close to 1 due to the rising speed and decreasing power consumption. In the second case, a lower reduction of CoT is also obtained, since the robot already achieved a medium speed (0.6 m/sec) before transition. In both cases, the CoT is improved by the affect of optimal joint stiffness.

Energy efficiency is one of the L-MESTRAN’s design features. The experimental results demonstrates that maintaining and changing stiffness by employing L-MESTRAN does not consume high power in comparison to the power to oscillate the leg. During both fixed stiffness and varying stiffness time, although the power consumption of the stiffness motor is much smaller than that of the hip motor, the efficiency/CoT of the hopping locomotion greatly improved. Moreover, the transition period in increasing stiffness is
Figure 2.14: Hopping behavior in varying joint stiffness. (a) joint stiffness is varied from low to optimal value; (b) joint stiffness is varied from high to optimal value; The gray bar indicates the transition period of the leg behavior during stiffness regulation.
longer than that in stiffness reduction. The reason is that stiffening the leg requires the stiffness motor to work against external load, whereas reducing stiffness can be beneficial by external load.

2.5 Discussions

We have shown our study of how to improve energy efficiency of a segmented leg hopping by adjusting the joint stiffness according to the actuating frequency. We started addressing the issue by constructing a numerical model of the leg. We carried out a number of experiments to investigate the influences of the joint stiffness in the CoT at different actuating frequencies. The simulation results revealed that at different actuating frequencies, there is always a joint stiffness for optimal CoT. We validated these results in a real robot platform. We also demonstrated the unique features of a novel variable stiffness mechanism in a series of experiments. There are several significant highlights of this paper derived from the simulation and experimental results as follows.

There are similarities between the simulation and experimental results not only in CoT but also in speed and power consumption. Here, the CoT, the speed, and the power consumption are examined. We have summarized five points from the simulation results, mentioned in the end of section 2. The relationship between the speed, the power consumption, and the CoT are interesting to be discussed as shown in Fig. 2.3, Fig. 2.4, Fig. 2.13, and Fig. 2.15. At all actuating frequencies, the speed is always a sensitive factor resulting in the variation of the CoT, and reaches maximum near the optimal joint stiffness. In contrast, the power consumption varies in a more monotonic way such that higher joint stiffness results in higher power consumption. It indicates that optimizing the speed is more effective for low CoT than power consumption. Indeed, when the speed is optimized, low power consumption is also achieved. Furthermore, minimum power consumption does not always guarantee maximum speed. The reason is the followings. When the stiffness is optimized, the natural frequency of the leg is close to its actuated frequency, as shown in Fig 2.3. The leg gains maximum elastic energy contributed by the compliant joint. Therefore, the power consumption is also significantly reduced. Apart from this optimal stiffness, when the joint stiffness is lower than the optimum, the motor can not exert high torques on the joint which leads to reduced power consumption and the ground reaction forces. Basically, the leg only oscillates in place and the energy input is dissipated in the compliant joint. When the joint stiffness is increased further from the optimal one, the natural releasing period of the
compliant joint is shorten. The leg does not only lose the elastic energy but also is disturbed by the its natural oscillation which is out of phase with the actuated oscillation from the motor during stance.

Optimizing energy efficiency via joint stiffness is an economical solution. At a given particular speed, humans can either shorten their stride length at high stepping frequencies or take a long stride length and reduce the stepping frequency. The experiments of human running at different speeds have shown that minimum energy consumption is achieved when the stepping frequencies are close to the optimum. When the speed is increased, these optimum frequencies also increased ([34], [35], [15], [16]). The results have indicated that biological systems have preferable stepping frequencies given a speed of locomotion. It is widely accepted that the muscle-tendon system in the human body can be regarded as a mass-spring-damper system in which tendons work as springs, muscles are the actuators, and friction in the joint provides the damping. The re-bouncing rate of this system under load is determined by its natural frequency. In order to run, walk, or hop efficiently, the stepping frequencies must be close to the natural frequencies to exploit the body dynamics ([16]). Inspired by biological studies, we investigate increasing energy efficiency by imposing the coupling between the stiffness and the actuating frequency. Thus, the amplitude $A$ and offset $q_{10}$ in the leg actuation, mentioned in Eq. 2.18 were heuristically chosen and fixed in all experiments. As a consequence, the minimum achievable $CoT$ is not necessarily a global minimum. The $CoT$ is possibly reduced further if $A$ and $q_{10}$ are also optimized. In order to comprehensively investigate the influences of the other parameters on the $CoT$, not only from joint stiffness and frequencies, it is required a systematic optimization over all control parameters. Indeed, enlarging the leg sweeping angle at a given actuating frequency can increase energy to move the leg. Furthermore, larger sweeping angle also induces larger center of mass excursion which costs more energy. Therefore, utilizing joint stiffness is still an economical solution to increase energy efficiency provided maintaining stiffness does not require external energy.

After a lot of experiments, the L-MESTRAN has demonstrated that its unique designed features enables the usability of the mechanism for improving energy efficiency highly dynamic hopping locomotion. In the first physical sense, the design of L-MESTRAN is small and compact allowing the mechanism to be integrated in a small, robust and lightweight leg platform. We were able to run the leg platform at a variety of frequencies (i.e. 2-6 Hz) which results in relatively high speed (i.e. 1.4 (m/s) or 5.6 leglength/sec). Although the controller is a simple, open loop and the leg only performs simple swings back and forth, the $CoT$ can still be reduced to 1.1. Second, L-MESTRAN can regulate the joint stiffness in a very large range, from zero
to infinity in principle. Utilizing this feature, the leg properties can be optimized to exploit its own dynamics at various actuating frequencies. This strengthens the scalability of the design since the range of the desired stiffness does not depend on the size of the mechanism. Furthermore, it is well understood that running in different surface stiffness also requires to regulate the leg stiffness ([36], [37], [38]). In fact, the surface stiffness can be varied from very soft like mud to very hard like concrete. Thus, the large range of reachable stiffness of L-MESTRAN can be exploited to accommodate for the surface stiffness change. Third, the L-MESTRAN design is capable to maintain the stiffness without external energy. This feature is not found of other variable stiffness actuators ([30], [31]) as well as biological actuators as muscles ([39]). Humans and animals in locomotion don’t often change stiffness, but rather other control parameters such as leg swing amplitude, offset angles below the moderate speed ([40], [41]). The reason is that the biological stiffness mechanism works in agonist-antagonist fashion which consumes energy for maintaining/stiffening the limbs even when the limbs don’t perform any mechanical works. In this aspect, the design of L-METRAN is more energy efficient than an equivalent biological tendon-muscle system. Forth, in terms of control, the L-MESTRAN can alleviate control effort since the joint stiffness is not changed by external load, therefore a simple PD control can be used to adjust the joint stiffness.

2.6 Conclusions

This paper studies the relationship between the joint stiffness and the actuating frequency to improve energy efficiency of hopping locomotion by using a planar one-leg robot. A numerical model as well as a physical platform have been developed and experimented. The experimental results have validated the finding from the simulation. We have found that energy efficiency of hopping locomotion in segmented leg can be greatly improved if the joint stiffness is appropriately adjusted to the actuating frequency. Thus, the leg robot is not constrained to predefined actuating frequencies during hopping for energy efficiency as long as the joint stiffness can be actively regulated. Furthermore, the optimal frequencies for the minimum CoT are close to the leg’s natural frequencies. This finding can be used to optimize the leg design for desired actuating frequencies. It can also advice the hopping controller to control the joint stiffness given an existing leg structure. In order to support this study, we have successfully developed a variable stiffness mechanism (L-MESTRAN) in the physical legged platform to carry out number of experiments. We have demonstrated that the mechanism is an effective
Figure 2.15: CoT, power consumption and speed of the real-world hopping robot at 2Hz (a), 3Hz (b), 5Hz(c),6Hz(d). For visualization purpose, the speed is scaled up by a factor of $mg$ where $m$ is the total mass of the robot and $g$ is a gravitational constant.
mechanism to improve energy efficiency in a segmented leg hopping.
Chapter 3

Development of Pneumatically Driven Quadruped Robot with Actuated Spine for Cheetah-like Behavior

3.1 Introduction

Prior research has concluded that the spine plays a critical role in quadrupedal locomotion [12]. A quadruped robot with an actuated spine named Kitty was developed, from which results supporting this claim were gathered. Kitty was limited by its size and mechanical structure, though, and was not able to demonstrate dynamic running. This paper describes the development of a biologically inspired quadruped robot, which will be used to further investigate the role the spine plays in quadrupedal locomotion. Renny, named after the German word ‘rennen’ which means to run, was developed with the goal for it to successfully walk and run in multiple gaits with minimal control complexity. In addition, Renny was also designed with a configurable body morphology, so that experiments could be run to see how changes in its structure, in particular its spinal morphology, impact locomotive performance.

In this report, the pneumatic actuation system used on the robot will first be introduced. Initial prototypes and sub-system testing of the robot will then be described. The robot design and its locomotive performance follows. Finally, opportunities for future work on the robot are described. Renny has been designed, built, and has successfully demonstrated running in a bounding gait. While additional work is yet to be done to quantify
the results, initial tests show that spine actuation allows Renny to achieve a longer stride length, allowing for faster locomotion.

### 3.2 Pneumatic Actuation System

McKibben-type pneumatic actuation was used on this robot. Because one of the main goals of Renny is to demonstrate biologically inspired quadruped locomotion, the inherent and tunable compliance and bio-mimetic characteristics of pneumatic actuation make it a better source of actuation than motor-driven actuation. McKibben actuators were chosen because of their success in prior legged robot applications. In [43], a quadruped robot with an actuated spine was able to achieve bounding locomotion using McKibben actuation. In [44], locomotion in multiple different gaits was achieved with the use of McKibben actuation.

Experiments were first run to gain experience fabricating and operating McKibben actuators. Figure 3.1 shows one of the actuators used on the robot. They were fabricated in the lab from scratch using 6 mm diameter hose, 6 mm diameter silicone rubber tubing, and 3/8 inch nominal diameter polyester braided-mesh sleeve. Plastic end-caps were used to seal the free end of the actuators and metal pipe clamps were used to secure the braided sleeve to the flexible tubing on each end of the actuator [45]. Throughout the remainder of the paper, actuators will be described by the length of inner tubing and nylon sleeve mounted on the actuator between the end of the hose connected to the pressure source and the end stopper at the end of the actuator. This is essentially the length of each of the two materials that is contributing to the actuation.

Experiments were run to measure how actuator performance changes as a function of certain actuator parameters. The first experiment run involved testing how varying the air supply pressure and actuator preload affect mus-
Figure 3.2: Plot of actuator deformation versus preload at different actuation pressures. Overlaid, in the scale shown on the right of the graph, is the depressurized length of the actuator at each preload.

Muscle contraction vs. Preload and Supply Pressure

Figure 3.2 shows the average actuator deformation measured using different supply pressures and actuator preloads. As you can see, deformation increases significantly as supply pressure is increased. This shows that supply pressure can be used as a control lever when tuning the robot’s gaits. Because actuators are more likely to fail at higher pressures, though, operation at or above 4 bar was avoided.

The data also shows that there is a consistent trend between actuator preload and deformation. The maximum deformation observed increased as the preload was increased up to a point where the actuator was nearly fully stretched while unpressurized. Beyond this preload, the deformation decreased. This shows that the actuator preload can also be a useful parameter to adjust to get an actuator to behave differently and alter the robot’s morphology.

The second actuator test performed investigated how changing the ratio...
of outer casing length to inner tube length impacts muscle deformation. This test used the same setup as described above, but now multiple actuators with different length ratios were tested over the same range of test conditions. A 2 bar supply pressure was used in all tests.

The results of this experiment are shown in Figure 3.3. All actuators whose data is plotted had an inner tube length of 5 cm. Additional tests were run with actuators with inner tube lengths of 4 and 6 cm to ensure that the trends observed were consistent. It was observed that increasing length ratio increases the deformation achieved. Additionally, it was observed that the preload at which the maximum deformation is achieved gets higher as the length of the outer casing grows. This makes sense, since achieving additional deformation requires the inner tubing, which behaves like a nonlinear spring, to stretch more and become stiffer. It was decided to use actuators with an outer casing to inner tubing length ratio of 2.25 on the robot. While more deformation could be achieved with a higher ratio, 2.25 ratio actuators demonstrated sufficient deformation for this application. Additionally, it was thought that using lower length ratio actuators would decrease the loads on the actuators over their life, making them fail less often.

While this testing was far from a complete characterization of the actuators’ behavior, enough was learned from the tests to move on to the robot design. Later in the robot development, it was observed that the way the actuators were tested was not consistent with the way the actuators are used on the robot. The tests were run with the mass hanging from the actuator serving as both the muscle preload and the mass resisting actuator deformation. Because of this, each actuator was only tested with one magnitude of
force resisting actuation at each preload (for example, at high preloads, the actuators were always resisting a large force, since it took a large mass to supply the large preload). It would be useful to run additional experiments in which the muscle preload and the resisting force were decoupled. This would make it possible to observe how the maximum actuator force transmitted changes with actuator preload. This would give data more representative of how the actuators behave on the robot, where the preload is independent of the force resisting actuation.

3.3 Prototypes and Early Testing

Two prototypes were fabricated to test sub-systems of the robot. First, a single joint prototype was fabricated to test the antagonistic muscle setup planned for use on the robot. Then, a prototype quadruped was fabricated with actuated joints located only at the four hips.

The first prototype can be seen in Figure 3.4. It consists of one hinge joint, intended to represent one of the hip joints of the robot. The joint mates an aluminum U-channel with a piece of 3D-printed acrylonitrile-butadiene-styrene (ABS) plastic, which is then connected to hollow square aluminum shaft. A 3 mm diameter brass shaft connects the two members of the joint through two standard ball bearings, mounted on the ABS side of the joint. Actuators are mounted on both the front and back of the joint. The free end of the actuators are mounted to an ABS-printed connector which can be slid up and down on the aluminum shaft and secured with screws. Adjusting this connector’s position allows the preload of the actuators to be adjusted, since its position determines the mounted length of the actuators, and hence the force in the actuators when unpressurized.

![Figure 3.4: Picture of the prototype leg during testing.](image)
Actuation is provided by one three-way proportional valve (part number MPYE-5-1/8 LF-010-B). With the valve commanded to one of its extreme positions, full actuation pressure is sent to the corresponding actuator and pressure is exhausted from the other actuator. With the valve in the center position, the pressure in both actuators is held constant. Pneumatic pressure is provided by an air compressor, with its output fed through a switch and pressure regulator, before being fed into the input of the control valve. A rough schematic of the pneumatic system is shown in Figure 3.5.

Some important observations were made from the testing of the first prototype joint. First, it was found that the antagonistically actuated joint could be actuated at up to about 3 Hz. At faster speeds, it appeared that the mechanical delays in the system started to dominate and the joint movement became inconsistent. 3 Hz is sufficiently fast for running in this application. It was also observed that, as expected, adjusting the muscle preload was effective at changing the joint’s unforced range of motion and actuation power.

Tests were run to see if different joint behaviors could be achieved with different valve control patterns. It was observed that virtually no difference in actuator pressure response could be achieved by decreasing the magnitude of the command sent to the valve. Additionally, ramping up the valve control command slowly over time did not have a noticeable effect on actuator behavior. This is most likely due to the fact that the valves used are designed for quite high flow rates and only a small amount of air is required to pressurize the actuators. As a result, barely opening the valve provides enough flow to pressurize the actuators almost immediately. This inability to modulate actuator pressure between zero and full pressure is a significant constraint and will most likely need to be remedied in future versions of the pneumatic system. For the time being, it was accepted that actuator pressure would
only be controlled via step inputs from zero to full pressure. Because one of the goals of this robot was to achieve robust locomotive capabilities with minimal control complexity, this was acceptable.

The second prototype built, shown in Figure 3.6, combined four joints similar to the one described above, to create a primitive quadruped robot with a stiff back and only one degree of freedom in each leg. Each joint had two antagonistically positioned actuators and each joint was powered by one three-way proportional valve, in the same way described above. The purpose for this prototype was simply to test the coordination between multiple control valves and ensure that the pneumatic system was working properly. This was confirmed and the prototype was able to demonstrate coordinated joint movement. It was not able to properly walk or run, but it did demonstrate a "moon-walk" behavior, where the robot stayed in the same place while its feet slid on the ground in a walking motion. This inability to walk was not surprising, since the robot did not have a means to achieve clearance between its feet and the ground, and it was completely symmetric from back-to-front. As a result, there was no mechanism to propel it in any particular direction.

3.4 Robot Design

Figure 3.7 (a) shows the joint and actuator configuration of the final robot and Figure 3.7 (b) shows the constructed robot. Renny has 2 actuated joints on each of its four legs. In addition, Renny has a spine which can be configured to be either stiff, passive with tunable stiffness, or actively actuated.
Figure 3.7: (a) Rough schematic of one side of the symmetric robot. Joints are shown in black, and actuators in yellow. (b) Picture of the completed robot.

Renny was designed such that its body morphology could be adjusted easily. This configurability provides the ability to easily test the impact of body morphology changes on locomotive performance.

3.4.1 Leg Design

All legs have two actuated degrees of freedom. For each leg, the hip is actuated by two McKibben actuators mounted horizontally above the hip joint, and the knee is actuated by two McKibben actuators mounted on the femur. At the end of each leg is a spherically shaped foot which is covered by a layer of foam to increase friction and decrease slippage. The range of motion of each hip is mechanically limited to +/- 45 degrees from center, and that of each knee is limited from 0 (corresponding to leg straight) to 80 degrees. Of course, the full range of motion is almost never achieved, since the actuators also limit the range of motion of each joint.

Because the goal of this robot is to investigate the role morphology plays in locomotion, the legs were designed such that their behavior could be easily adjusted without the need for additional hardware. In Figure 3.8 (a) you can see where the actuators are mounted on the leg. In the hips, the air supply end of the actuators are mounted to the two holes above and below the hip joint. The free ends of the actuators are mounted to the rectangular connector in the upper right of the image, which can slide along the horizontally mounted aluminum channel. By increasing this connector’s distance from the hip joint, the joint can be stiffened and its range of motion can be decreased. Multiple attachment holes are provided for each actuator on the sliding connector as well, to allow the antagonistic actuators to have different mounted lengths. This gives the ability to offset the at-rest position of the
Figure 3.8: (a) CAD screen-shot of Renny’s rear legs. The holes to which the actuators are fastened to the robot are highlighted in red and blue, with the red holes signifying those which allow the mounted length of the actuators to be adjusted. The holes to which the hip’s muscle connector piece can be fastened are highlighted in yellow. (b) CAD drawing of one of Renny’s knee joints, with mating surfaces for the bearing and inner shaft highlighted in red.

It also gives the ability to limit the range of motion of the joint in one desired direction. For example, if you wanted to limit the amount a rear hip joint could kick the leg back, you could mount the actuator above the hip joint such that it is nearly fully stretched. This limits how far back the leg can swing, regardless of how hard the opposite muscle pulls. This same adjustability is provided in the knee joints as well. For the knees, the mounted length of the actuators is adjusted by changing the holes in which the air supply ends of the actuators (the end closest to the hip joint) are mounted. All actuators used in the hip joints use an inner tube length of 5 cm and outer casing length of 11.25 cm. All actuators in the knee joints use an inner tube length of 4 cm and outer casing length of 9 cm.

All joints in the leg are made up of two ABS 3D-printed parts; each of which is attached to a 1 cm square hollow aluminum rod. For each joint, the two printed parts are connected by a 3 mm diameter bronze shaft and one 4 mm wide ball bearing. Previous prototypes used two bearings in each joint, but it was found that one bearing could handle the loads required during running. The final one-bearing design was much easier to assemble and resulted in lower and more consistent friction within the joints. Each joint has an additional flange, which can be seen on the left of Figure 3.8 (b),
on which a potentiometer can be mounted to allow for the measurement of joint angles during testing.

While the leg joints have had no hardware failures and have worked well throughout testing, there is some room for improvement in the leg design. The bearings used in all leg joints are standard ball bearings. Because the joints may experience significant lateral forces during running in more complex gaits and on uneven terrain, it may be necessary to replace these bearing with some which can better handle the lateral twisting loads. Cylindrical or needle roller bearings may be more appropriate for this application. The length and at-rest positions of the legs may also provide room for improvement. The initial design was chosen to give proportions similar to that of a medium sized dog, but no tests were run to analyze how changing these parameters impacts performance.

3.4.2 Spine Design

Because little was known about the optimal muscular configuration in the spine for a quadruped like the one designed, Renny’s spine was designed for full configurability. The final design is shown in Figure 3.9 (a). The spine consists of one ball and socket joint, with a range of motion of 20 degrees from center in all directions. The joint is made up of two ABS printed parts, designed such that there is flat surface contact when any extreme of the range of motion is reached. To keep the ”vertebrae” from dislocating, a metal wire running through the center line of the two parts holds them together. Two circular ABS discs, mounted on either side of the spinal joint, allow up to eight pneumatic actuators to be mounted in various configurations around the spine. The distance between the discs can be adjusted, as well as where the center point between the disk lies in relation to the spinal joint. Each actuator is mounted to the discs by two steel C-hooks. These hooks can be screwed into and out of the discs to adjust the mounted length of each individual actuator.

In the initial spine design, only one ball and socket joint was used to simplify joint motion, make mechanical binding less likely, and to make it easier to analyze the body dynamics during testing. In the future, as more is learned about the optimal spinal structure, more vertebrae can be added to the spine to give it more degrees of freedom and a more biologically similar morphology.
3.4.3 Control Design

Actuation for each pair of antagonistic muscles is provided by one 3-position proportional valve, in the configuration described in Section 3. For initial gait development and testing, a very basic sequential step-function control pattern was used. Each control move was timed as a percentage of the gait period. This allowed multiple gait periods to be easily tested for each gait. All control was completely open-loop and hand-tuned. Both a bounding and trotting gait were designed and tested.

Figure 3.10 shows the control sequences developed for the bounding and trotting gaits. In both cases, the gaits were developed using a dog’s movement as the desired behavior. In the bounding gait, control action is symmetric from left to right, so the commands for only two legs are shown. In the trotting gait, no testing has been done with spine actuation, so its control signal is not shown.

3.5 Results

Renny successfully demonstrated bounding both with and without spine actuation. Continuous bounding was achieved at 2, 3, and 4 Hz. In all bounding tests, three actuators were used in the spine, one centered above the spine, and two separated by about 8 cm below the spine. With no spine actuation, all spinal actuators were kept at 3.5 bar, making the spinal joint quite stiff but still mildly compliant. With spine actuation, the actuator above the spine was held at high pressure, behaving like a stiff spring, while the two
Figure 3.10: Control actuation sequences for the bounding (a) and trotting (b) gaits.

Actuators below the spine were tensed and relaxed in accordance with the control signal. This three-actuator morphology was found through trial and error. It was observed that the spinal joint needs to be quite stiff at all times to maintain stability, which is why the top actuator was held at full pressure even when the spine was being actuated. More spinal actuators will certainly be needed to allow for more complex spinal actuation in different gaits, but a minimal number of actuators were used during gait development to limit the amount of control parameters to be explored during tuning.

Figure 3.11 shows a sequence of images of Renny while bounding both with and without spine actuation. In both configurations, a short phase in which all feet were off the ground was observed. You can see that body extension and contraction is much more pronounced with spinal actuation. This allows Renny to make longer strides. While insufficient measurements were taken to confirm this, the longer strides should allow Renny to run faster when its spine is being actuated.

During testing, it was observed that the stiffness of the spinal actuators has a significant impact on Renny’s stability during running. At one point during tuning, Renny completely lost the ability to run and was falling over in every trial. Shortly after, it was observed that one of the C-hooks on which the actuator above the spine was mounted had loosened, making that actuator less stiff. Once the C-hook was returned to its original position, Renny’s ability to run was restored. It would be useful to run a similar series of tests, in which the stiffness of the spinal actuators is deliberately decreased to measure the stiffness at which locomotive capability is lost.

Renny has not successfully trotted using the purely open-loop control strategy. After a couple of steps, one foot seems to get stuck in the forward-most position due to poor weight distribution. That leg then falls out of sync.
with the others, and in the next couple of steps, Renny falls over. It seems like this behavior is the result of small differences in the actuation behavior of each leg. More sophisticated control is most likely needed to achieve trotting.

### 3.6 Future Work

There are many opportunities for future work on Renny. Most importantly, additional work needs to be done to quantify the findings of the bounding gait testing. By equipping Renny with joint angle sensors and using video key point tracking to analyze its dynamics, it should be possible to show that Renny can run faster with spine actuation than without. The timing and power of spine actuation may need to be further refined to achieve this. It would be interesting to try using ground contact sensing to close the loop in the control sequence and better time the actuation of the spine. It would also be interesting to investigate how spine actuation affects Renny’s pneumatic energy consumption during running. It may be possible to show that the actuation pressure required in the legs decreases when the spine is actuated.

There is a lot of room for improvement in the pneumatic system used for this robot. Currently, Renny is connected to each control valve by a 6 mm diameter piece of tubing. As a result, Renny’s tether is quite large and weighs more than he does. This makes it very difficult to get consistent results during testing and makes it easy to unintentionally offset Renny’s center of gravity. It may be possible to use smaller multi-channel tubing that is lighter and more compact. As an alternative, it would be even better to mount small control valves directly to the robot.
More sophisticated actuation pressure control is most likely required for Renny to achieve more complicated gaits. Currently, Renny cannot walk because the actuation pressure control does not allow Renny’s legs to kick back much slower than they are swung forward, as is required for walking. This is because the same actuation pressure is used for the extensor and flexor actuation. It should be investigated what can be achieved by making the extensor and flexor actuation independent of each other. Can more sophisticated joint movements be achieved by staggering when the extensor and flexor of each joint is pressurized and depressurized? Can walking be achieved simply by supplying the actuators which kick the legs back with less pressure than those that swing the leg forward?

As an additional level of complexity, it may also be possible to use pressure sensing within each actuator to perform closed-loop pressure control on each actuator and actively modulate actuator pressure. If this is possible, can one then gain full control over each joint’s trajectory? If so, quite sophisticated gaits and very complex behavior should be achievable.

3.7 Conclusion

Renny has successfully demonstrated running with and without spine actuation. Initial results suggest that spinal actuation does improve performance in the bounding gate, but additional testing needs to be done to prove this quantitatively. The robotic platform developed should serve as a useful tool for further investigating the effect of body morphology on quadrupedal locomotion performance.
Chapter 4

Covariation of Body Parameters Compensates Mass Disturbances in Human Locomotion

4.1 Introduction

The study of bipedal locomotion has motivated the development of conceptual models to explain the most relevant principles ruling the dynamics of a gait \([46]\). The spring-loaded inverted pendulum (SLIP)\((47)\) is one of this models that has been used extensively to explain the running gait. Geyer et al.\((48)\) proposed the SLIP model as a unifying framework to describe walking as well by adding an extra massless spring representing the second leg.

This unified perspective explains the exchange of kinetic and potential energy of the center of mass (CoM) for walking and running in human bipedal locomotion. In running the kinetic and potential energy of the CoM are in phase, while during walking the 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\((49)\). In addition, the model correctly reproduces the vertical oscillations of the CoM observed experimentally \((48, 50)\). Furthermore, the SLIP model can show gait transitions at constant energy as it is reported in Ref. \([51]\). Based on this findings, Martinez et al.\((52)\) showed that these transitions enable the system to generate compound gaits like hopping which is a combination of running and walking. The explanatory capabilities of this model have been exploited in the area of robotics as well.
In this field, the model has been successfully used for design and control of running machines (53–56).

The SLIP model consists of a point mass (the body) attached to a massless spring (the leg) (Fig. 4.1). The angle of attack ($\alpha$) is the angle spanned by the landing leg and the horizontal at the time the foot collides with the ground. Given that the legs in the SLIP model are massless, their swinging motion is not described by the Newton’s laws. For this reason, a control policy sets this angle at touchdown. In this study, we assume that the angle of attack at touchdown is kept constant.

![Figure 4.1](image)

Figure 4.1: (Color online) Illustration of the evolution of the SLIP model for running and walking. The different phases are indicated. Due to the passive properties of these models, control is necessary only during the swing of the leg.

In this research, we analyze the effect of variations of mass in the human gait under the condition of constant locomotion speed. We account for subject differences in body mass and GRF with a non-dimensional analysis technique. First, we fitted the experimental data with the SLIP model. Second, we derived three dimensionless variables from the parameters of the SLIP model. And third, we compared the changes on the these variables as a function of the change in the mass. Results from this study show that changes on the mass are compensated within the subject leading to a similar dimensionless system.

This paper is organized as follows. In section 4.2, we describe the experiments, we define the models used for data analysis, and the algorithm used to find the correct model parameters to fit the experimental data. In section 4.3, we show the data collected in the experiments, the accuracy of the model fitting, and the changes in the non-dimensional numbers as a function of the mass variations. Later, in section 4.4, we discuss about the relations in the
model with the change of mass and the implications for locomotion control. We conclude the paper in section 4.5 with our conclusion.

4.2 Materials and methods

In this section we introduce the experiments, the model that we used to analyze the data collected, and the procedure that we used to fit the model to the experimental data.

4.2.1 Experiment

We measured kinetic data using a split-belt force treadmill. The force plates underneath the frame measured ground reaction forces (GRF) and moments in three axes, at a sampling rate of 960 Hz. Four subjects (three male and one female subjects; Body mass 65.34 kg SD 7.07 kg) signed informed consent forms approved by the University of Michigan Health Sciences Institutional Review Board, and participated in this study. All subjects were considered to be in good health, and had no known gait abnormalities.

For all subjects, we collected data during three treadmill walking and running trials. In the walking condition the speed was held constant at 1.5 m/s for all trials. In the first trial all subjects walked normally in the treadmill for 60 s. In the second trial all subjects walked with a weighted belt with 9.3 kg for 60 s. In the third trial the all subjects walked with a weighted belt and a weighted vest with 20.5 kg for 60 s. In the running condition the speed was held constant at 2.5 m/s for all trials. In the running experiment we followed the same mass variations and locomotion time used in the walking experiment. Subjects were given rest periods of up to 3 minutes between trials in both conditions.

4.2.2 Model

As explained previously, we use the SLIP model to study the effect of increments of mass in bipedal locomotion. A complete description with full details can be obtained in Ref. [51]; herein we provide a succinct recapitulation. The model represents the different phases of a gait with three sub-models or phases. Each phase represents 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 represents the body and the massless linear springs model the forces from the legs. We define a running gait as a trajectory that switches from the
single stance phase to the flight phase and back to the single stance phase. A walking gait is defined as a trajectory that switches from the single stance phase to the double stance phase and back again to the single stance phase.

The equation of motion of the flight phase is described in Cartesian coordinates as a parabolic motion in which the system is affected only by the gravitational field.

\[
\begin{pmatrix}
\ddot{x} \\
\ddot{y}
\end{pmatrix} = \begin{pmatrix}
0 \\
-g
\end{pmatrix},
\]

where \( \ddot{x} \) is the acceleration of the center of mass in the horizontal direction, \( \ddot{y} \) is the acceleration of the center of mass in the vertical direction, and \( g \) is the acceleration due to gravity.

We can write also the equation of motion of the single stance phase in Cartesian coordinates.

\[
\begin{pmatrix}
\ddot{x} \\
\ddot{y}
\end{pmatrix} = \begin{pmatrix}
\frac{k}{m} (r_0 - r_1) \cos \theta \\
\frac{k}{m} (r_0 - r_1) \sin \theta - g
\end{pmatrix},
\]

where \( m \) is the mass, \( k \) is the constant of elasticity of the spring, \( g \) is gravity, \( r_0 \) is the natural length of the spring, \( r_1 \) is the length of the spring, and \( \theta \) is the angle spanned by the leg and the horizontal, growing in clockwise direction.

The equation of motion of the double stance phase in Cartesian coordinates is

\[
\begin{pmatrix}
\ddot{x} \\
\ddot{y}
\end{pmatrix} = \begin{pmatrix}
\frac{k}{m} (r_0 - r_1) \cos \theta + \frac{k}{m} (r_0 - r_2) \cos \beta \\
\frac{k}{m} (r_0 - r_1) \sin \theta + \frac{k}{m} (r_0 - r_2) \sin \beta - g
\end{pmatrix},
\]

where \( m \) is the mass, \( k \) is the constant of elasticity of the springs, \( g \) is gravity, \( r_0 \) is the natural length of the springs, \( r_1 \) is the length of the spring of the rear leg, \( \theta \) is the angle spanned by the rear leg and the horizontal (growing in clockwise direction), \( r_2 \) is the length of the spring of the hind leg, \( \beta \) is the angle spanned by the hind leg and the horizontal (growing in clockwise direction).

We can define the non-dimensional distance \( \hat{l} \) as \( \hat{l} = l/r_0 \), where \( l \) is a dimensional distance and \( r_0 \) is the natural length of the spring, and the dimensionless time \( \hat{t} \) as \( \hat{t} = t \sqrt{g/r_0} \), where \( t \) is time, \( g \) is gravity and \( r_0 \) is the natural length of the spring. Using this relations we can convert the velocities (\( \dot{\hat{l}} = \dot{l}/\sqrt{g/r_0} \)), and accelerations (\( \ddot{\hat{l}} = \ddot{l}/g \)) to the non-dimensional space. With these definitions we can rewrite all the previous differential equations in the dimensionless space. In the case of the flight phase the equation of motion is:
\[ \begin{pmatrix} \ddot{x} \\ \ddot{y} \end{pmatrix} = \begin{pmatrix} 0 \\ -1 \end{pmatrix}, \quad (4.4) \]

where \( \ddot{y} \) is the non-dimensional acceleration of the center of mass in the horizontal direction, and \( \ddot{y} \) is the non-dimensional acceleration of the center of mass in the vertical direction.

The equation of motion of the single support phase in the non-dimensional space is equal to:

\[ \begin{pmatrix} \ddot{x} \\ \ddot{y} \end{pmatrix} = \begin{pmatrix} \hat{k} (1 - \hat{r}_1) \cos \theta \\ \hat{k} (1 - \hat{r}_1) \sin \theta - 1 \end{pmatrix}, \quad (4.5) \]

\[ \hat{k} = \frac{kr_0}{gm}, \quad (4.6) \]

where \( \hat{k} \) is the non-dimensional stiffness of the spring, and \( \hat{r}_1 \) is the dimensionless length of the leg.

In a similar way we can rewrite the equation of motion of the double stance phase as:

\[ \begin{pmatrix} \ddot{x} \\ \ddot{y} \end{pmatrix} = \begin{pmatrix} \hat{k} (1 - \hat{r}_1) \cos \theta + \hat{k} (1 - \hat{r}_2) \cos \beta \\ \hat{k} (1 - \hat{r}_1) \sin \theta + \hat{k} (1 - \hat{r}_2) \sin \beta - 1 \end{pmatrix}, \quad (4.7) \]

\[ \hat{k} = \frac{kr_0}{gm}, \quad (4.8) \]

where \( \hat{k} \) is the non-dimensional stiffness of the springs, \( \hat{r}_1 \) is the dimensionless length of the rear leg, and \( \hat{r}_2 \) is the non-dimensional length of the hind leg.

One important result from the reformulated system is that single stance phase and the double stance phase do not require another dimensionless number. therefore, we need only three numbers (non-dimensional stiffness \( \hat{k} = \frac{kr_0}{gm} \), non-dimensional time \( \hat{t} = t \sqrt{g/r_0} \), and non-dimensional length \( \hat{l} = l/r_0 \)) to represent any gait based on the SLIP model. The Combination of these three numbers allow us to convert other important physical quantities to the non-dimensional space e.g. energy, force, velocities, etc.

### 4.2.3 Model Fits

The model implementation and data analysis were performed in MATLAB(2009, The MathWorks). We used the MATLAB fminsearch function to fit the SLIP model to the experimental GRF. We selected the squared error between the
experimental GRF and the GRF generated by the model as a quality measure of the fitting. This measure considered the vertical and the horizontal GRF, as shown in Eq. 4.9

$$error = \sqrt{\sum_{i=0}^{n} [\Delta t(f_x(t_i) - \hat{f}_x(t_i))]^2 + [\Delta t(f_y(t) - \hat{f}_y(t_i))]^2}, \hspace{1cm} (4.9)$$

where $f_x(t_i)$ is the measurement of the total horizontal force at time $t_i$, $f_y(t_i)$ is the measurement of the total vertical force at time $t_i$, $\hat{f}_x(t_i)$ is the model approximation of the total horizontal force at time $t_i$, $\hat{f}_y(t_i)$ is the model approximation of the total vertical force at time $t_i$, and $\Delta t$ is the time between samples (e.g. $t_{i+1} - t_i$). The quality measure is calculated through a stride, this means that the leading foot touches the ground at $t_0$, and again at $t_n$.

The function fminsearch needed a initial guess trajectory to select the appropriate SLIP model parameters. This guess was selected from all the possible symmetric gaits that can be generated from the SLIP model with parameters $k = 15 \text{ kNm}$, $m = 80 \text{ kg}$, $r_0 = 1 \text{ m}$, and $g = 9.81 \text{ m/s}^2$ between 780 J and 2000 J. The selection was performed based on two measures. One was the duty factor (the fraction of the total duration of a gait cycle in which a given foot is on the ground), and the other was the ratio between the peak vertical force and the peak horizontal force. The trajectory which produced the closest values compared to the experimental data was selected as initial guess for the fitting process. Herein, the mass which was measured from the vertical force, and gravity are maintained constant.

### 4.3 Results

In this section, we introduce the results of the experiments and the fitting parameters of the SLIP model. Fig. 4.2 shows the GRF of all the subjects for the walking and running conditions. The second row of Fig. 4.2 shows the mean GRF in the non-dimensional space for the walking trials. In the non-dimensional space the GRF of each subject are similar, there is only a comparable difference in the third trial of the subject three. In this case the walking cycle takes less time than before. The fourth row of Fig. 4.2 shows the mean GRF in the non-dimensional space for running trials. For all the subjects, the bigger the mass the shorter the flight phase. For this reason, the duty factor increases as a function of the weight. Furthermore, the dimensionless vertical GRF magnitude changes as a function of the mass. The bigger the mass the smaller the peak force.
Figure 4.2: (Color online) GRF for all the subjects. Each column shows the data of each subject. The light gray (cyan) shows the data collected without additional weight. The gray (magenta) shows the data collected with the weighted belt, and the dark gray shows the data collected with the weighted belt and the weighted vest. The dashed lines depict the vertical GRF and the continuous the horizontal GRF. From top to bottom, the first row shows the data collected in the walking experiment. The standard deviation of the measurements are depicted in the shadow area. The second row shows the mean total vertical and horizontal GRF of each subject in the non-dimensional space for the walking experiment. The third row shows the data collected in the running experiment, and the fourth row shows the mean total vertical and horizontal GRF of each subject in the non-dimensional space for the running experiment.

The table shows the appropriate parameters of the model that best fit the experimental data. The column error in the table indicates the the mismatch between the experiment and the model as a percentage of the area.
of the experimental GRF. The column $\alpha$ in the table shows the angle of attack that is selected each step by the model. The column $\sqrt{r_0/g}$ shows the constant that converts the non-dimensional time to seconds. The column $\frac{kr_0}{gm}$ depicts the dimensional stiffness of the system. All these quantities are generated from the fitting procedure. The column mass indicates the mass value estimated from the integration of the vertical force, and the column gait shows the locomotion pattern used in the experiment. The fitting procedure shows that the model can represent the experimental data with an error lower than 1%. The change in the mass for all the subjects through the experiments is around 33%, however the angle of attack (control strategy) changes less than 2.67% in walking and less than 4.58% in running.

Table 4.1: Results fitting the data with the SLIP model

<table>
<thead>
<tr>
<th>Gait</th>
<th>Subject</th>
<th>Mass</th>
<th>$kr_0/mg$</th>
<th>$\sqrt{r_0/g}$</th>
<th>$\alpha$</th>
<th>$max\Delta\alpha(%)$</th>
<th>error(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Walking</td>
<td>subject 1</td>
<td>62.62</td>
<td>17.46</td>
<td>0.22</td>
<td>64.37</td>
<td>0.70</td>
<td>0.42</td>
</tr>
<tr>
<td></td>
<td></td>
<td>71.68</td>
<td>17.33</td>
<td>0.22</td>
<td>64.06</td>
<td>0.34</td>
<td>0.33</td>
</tr>
<tr>
<td></td>
<td></td>
<td>82.87</td>
<td>17.21</td>
<td>0.23</td>
<td>63.92</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>subject 2</td>
<td>56.64</td>
<td>18.26</td>
<td>0.26</td>
<td>63.51</td>
<td>0.60</td>
<td>0.51</td>
</tr>
<tr>
<td></td>
<td></td>
<td>66.01</td>
<td>18.13</td>
<td>0.26</td>
<td>63.19</td>
<td>0.44</td>
<td>0.41</td>
</tr>
<tr>
<td></td>
<td></td>
<td>77.00</td>
<td>17.88</td>
<td>0.26</td>
<td>63.13</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>subject 3</td>
<td>70.05</td>
<td>15.70</td>
<td>0.24</td>
<td>60.60</td>
<td>2.67</td>
<td>0.72</td>
</tr>
<tr>
<td></td>
<td></td>
<td>79.37</td>
<td>16.21</td>
<td>0.24</td>
<td>61.74</td>
<td>0.58</td>
<td>0.46</td>
</tr>
<tr>
<td></td>
<td></td>
<td>90.63</td>
<td>15.51</td>
<td>0.22</td>
<td>62.22</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>subject 4</td>
<td>72.03</td>
<td>17.00</td>
<td>0.25</td>
<td>62.84</td>
<td>0.46</td>
<td>0.48</td>
</tr>
<tr>
<td></td>
<td></td>
<td>81.34</td>
<td>16.96</td>
<td>0.25</td>
<td>63.13</td>
<td>0.43</td>
<td>0.39</td>
</tr>
<tr>
<td></td>
<td></td>
<td>92.73</td>
<td>16.90</td>
<td>0.25</td>
<td>63.09</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Running</td>
<td>subject 1</td>
<td>62.36</td>
<td>15.28</td>
<td>0.43</td>
<td>62.90</td>
<td>4.17</td>
<td>0.59</td>
</tr>
<tr>
<td></td>
<td></td>
<td>71.83</td>
<td>12.83</td>
<td>0.46</td>
<td>60.57</td>
<td>0.56</td>
<td>0.57</td>
</tr>
<tr>
<td></td>
<td></td>
<td>82.85</td>
<td>11.99</td>
<td>0.48</td>
<td>60.38</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>subject 2</td>
<td>56.27</td>
<td>15.41</td>
<td>0.43</td>
<td>65.24</td>
<td>4.17</td>
<td>0.66</td>
</tr>
<tr>
<td></td>
<td></td>
<td>65.94</td>
<td>13.93</td>
<td>0.44</td>
<td>63.65</td>
<td>0.59</td>
<td>0.63</td>
</tr>
<tr>
<td></td>
<td></td>
<td>77.26</td>
<td>13.05</td>
<td>0.45</td>
<td>62.63</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>subject 3</td>
<td>70.24</td>
<td>17.30</td>
<td>0.42</td>
<td>63.11</td>
<td>2.94</td>
<td>0.57</td>
</tr>
<tr>
<td></td>
<td></td>
<td>79.29</td>
<td>15.87</td>
<td>0.51</td>
<td>61.53</td>
<td>0.54</td>
<td>0.60</td>
</tr>
<tr>
<td></td>
<td></td>
<td>90.59</td>
<td>15.50</td>
<td>0.53</td>
<td>63.34</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>subject 4</td>
<td>72.22</td>
<td>14.81</td>
<td>0.44</td>
<td>62.54</td>
<td>4.58</td>
<td>0.82</td>
</tr>
<tr>
<td></td>
<td></td>
<td>81.52</td>
<td>13.48</td>
<td>0.48</td>
<td>61.09</td>
<td>0.80</td>
<td>0.90</td>
</tr>
<tr>
<td></td>
<td></td>
<td>92.71</td>
<td>12.45</td>
<td>0.47</td>
<td>59.80</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Fig. 4.3 shows an example of the model fitting to the experimental data in subject two. The fitting procedure assumes a symmetric gait, for this reason the same angle of attack is used for both legs. The experimental data of subject two shows that this assumption is close to the reality even when the human locomotion can be as in this case clearly asymmetric. In addition, in the running gait, the vertical GRF sometimes has a notch, however, despite these source of error, the SLIP model can reproduce the experimental data with an average error of 0.45% for walking and 0.63% for running.

Figure 4.3: (Color online) Comparison between the experimental GRF and the GRF generated with the SLIP model. the data collected from subject two in the walking and running experiment is shown in gray (magenta). The light gray (cyan) shows the model approximation. The dashed lines depict the vertical GRF and the continuous lines depict horizontal GRF. From top to bottom, the first row shows the walking experiment, and the second row shows the running experiment. The most left panels show the data collected without extra weight, the panels in the center show the data collected with the weighted belt, and the most right panels show the data collected with the weighted belt and the weighted vest.

We use the parameters from the fitting procedure to measure the similarity between the experimental trials. To analyze the results across all subjects,
we compare the changes of the dimensionless SLIP parameters against the change of mass. The left panel of Fig. 4.4 shows the linear regression of the change of non-dimensional stiffness as a function of the change of mass. This regression is very close to a horizontal line which tells us that the number \( \frac{k_{r0}}{gm} \) is maintained constant even when there are increments in the mass \( m \). The coefficient of determination of this regression \( R^2 \) is equal to 0.26. This coefficient compares the error obtained from the regression against a horizontal line going through the mean of all Y values. A value of \( R^2 \) close to zero indicates that knowing X does not help you predict Y. This confirms the idea that the non-dimensional stiffness does not change.

Figure 4.4: (Color online) Comparison between changes in the non-dimensional stiffness and changes in the mass. The light gray (cyan) markers show the change of the non-dimensional stiffness. The light gray (cyan) line is the linear regression that fits the change of the non-dimensional stiffness as a function of the change of the mass. The left panel shows the data comparison for the walking experiment. In this panel, the gray (magenta) markers show the change of the stiffness, and the gray (magenta) line is the linear regression that fits the change of the stiffness as a function of the change of the mass. The right panel shows the data comparison for the running condition. In this panel, the gray (magenta) markers show the change of the natural length.

Additionally, in general the stiffness compensate for the changes on the mass. The linear regression shows dependency between the change of the constant of elasticity with the change of the mass. This relation has a slope of 0.97 which is very close to one. This means that the constant of elasticity grows almost in the same proportion as the mass. However, the compensation can also be produced by the joint action of the stiffness and natural length.
The last trial of subject three shows this behavior. The SLIP model that fits the data of the trial with the biggest mass shows a particularly high stiffness (left panel Fig. 4.4), this increment in the stiffness is compensated by a reduction of the natural length of the leg. The natural length is used to define all the dimensionless numbers included the non-dimensional time. For this reason a decrement in the natural length of the leg will increase the stepping frequency. This explains why the subject three in the last trail reduces the walking period.

The right panel of Fig. 4.4 shows that the non-dimensional stiffness decreases. The linear regression shows a slope of $-0.49$ between the change of the dimensionless stiffness and the change of the mass. This implies, that there is a compensation because the slope is not equal to minus one, but this compensation is not as strong as in walking. Furthermore, the compensation is not carried out by the stiffness but by an increment in the natural length of the leg. The stiffness in the running trials do not increase, in the contrary it is reduced when the subject has a bigger mass. The combination of these two effects, on one hand decrease the stepping frequency, and on the other hand increases the contact time of the foot with the ground. For this reason, the increment of mass in running increase the duty factor (Fig. 4.2).

### 4.4 Discussion

This model has proven its capability to represent many different features of human locomotion, for this reason is not surprising that it approximates very well the experimental data. The approximation errors are small, however in the running experiment the approximation error is bigger than the one in walking (table 4.1). This can happen because the SLIP model assumes a spring with a linear constant of elasticity. However, experimental studies have shown that human legs have a nonlinear constant of elasticity [57, 58]. In the case of walking, the compression of the leg is smaller than in running. This may help to reduce the nonlinear effect making the model closer to the experimental data.

The simplicity of the SLIP model allow us to rewrite it with three different non-dimensional numbers. The dimensionless analysis allows a more general study of the human locomotion based on the relations of a small set of variables that describe the dynamics. Under this representation we can study how a change of one of these variables can affect the locomotion. In this study, we focus our attention in how the change in mass affects the human gait in the case of constant speed of locomotion. We identified for walking and running a compensation of the change of mass that tries to keep
the non-dimensional stiffness constant (Fig. 4.4). In the case of walking, the mechanism is to increase the stiffness of the leg. With this strategy, the dimensionless stiffness is practically constant. The compensation in running uses the natural length of the leg instead of the stiffness. We speculate that, this mechanism can be better than increasing the stiffness because in running the knee bends more, and an increment in the stiffness can require more effort.

We identified also in one trial a compensation carried out by the relation between the stiffness and the natural length of the leg. We speculate that this behavior can be plausible in a subject that prefers to walk in her toes. In this condition, probably when we increase drastically the mass there is a point in which the subject will use the whole foot as a support decreasing the natural length of the leg and increasing the constant of elasticity of the leg.

One consequence of keeping the same non-dimensional constant of elasticity in the SLIP model is that the strategy of control (constant angle of attack) does not have to change. This idea is supported by the results of the fitting process. In table 4.1 we can see that even when the change of the mass is around 33% the constant angle of attack changes less than 4%. This is true even for the running experiment, in which the compensation of the change of the mass is about 50%.

Non-dimensional analysis using the SLIP model allow us to compare the relevant variables that describe the dynamics of human locomotion. Furthermore, we can analyze across several subjects. However, we have to remember that this is an abstraction and that the natural length \( r_0 \) is not the length of the leg but the natural length of the equivalent spring that represents the leg. The same happens with the angle of attack which is not the same as the angle between the human leg and the ground, but between the equivalent spring and the ground. We believe that similar analysis to the one presented in this study can be developed to understand the control strategy adopted in human locomotion in the presence of other kind of disturbance.

4.5 Conclusion

We used the SLIP model to represent human gaits because it endows the analysis with a more general representation based on three dimensionless numbers: non-dimensional stiffness \( \hat{k} = \frac{k r_0}{g m} \), non-dimensional time \( \hat{t} = t \sqrt{\frac{g}{r_0}} \), and non-dimensional length \( \hat{l} = l/r_0 \). The Combination of these quantities allow us to convert other important physical quantities to the dimensionless space like energy or force. We studied how the change of mass affects the
human gait keeping the speed of locomotion constant. We identified that in human locomotion there is a compensation of the change of the mass that can be explained in terms of the dimensionless SLIP model numbers. A direct consequence of the mass compensation is that the control strategy of the gaits does not have to change. We strongly believe that this analysis can be extended to study other important aspects in the human gait.
Chapter 5

Reduced Dimensionality and Morphological Computation

5.1 Introduction

One of the issues that a growing creature has to contend with during development is the manner of acquisition of motor skills. In particular, the necessity to coordinate a large number of degrees of freedom in order to perform tasks places a significant burden on the learning and optimisation that needs to take place - the so-called curse of dimensionality. One strategy alleviating this problem, is to acquire reduced dimensional models which are then utilised internally for optimisation and control of behaviour; this has biological basis in the form of the theory of biological motor primitives and pattern based control. Reduced dimensional control has been hypothesised to be an important feature in biological systems and is usually manifested in the form of Motor Primitives [59, 60].

An extension of the work, an the implementation in a real tendon-driven compliant robot system that utilises series-elastic actuation is currently underway. Building upon this work, are explorations in two directions in understanding and quantifying the role of reduced dimensionality in motor skill acquisition of high dimensional/complex robot, followed by a case studies in a more biological context of oculomotor control.

The first direction was to understand the relationship of morphology and material properties (in the form of compliance - stiffness and damping) to reduced dimensionality [61]. A key result in this regard was to demonstrate that dimensionality is strongly affected by damping acting on physical elements. An extension of this preliminary work, which utilised spring-mass-chains, to compliant kinematic chains (i.e. arms) is currently underway. The second
direction is to examine the contribution of pattern based control input, decomposable to a set of basis functions, towards reduced dimensionality; this formulation for task-specific dimensionality reduction allows quantification of individual tasks as well as their optimization, on the basis of reduced dimensionality (papers under preparation). An important result in this regard is to quantify the dimensionality cost of different trajectories and in particular, develop a measure for incorporating waypoints into a trajectory. The overall scheme is lastly extended to a more biological context in a case study on oculomotor control. A model for ontogenetic acquisition of reduced dimensional control for horizontal eye movements is presented, that is compared against an existing model of minimum-norm control in eye-movement literature (paper under preparation).

5.2 Effect of Material Properties on Reduced Dimensionality

While it is well recognised that the physical phenomenon of the growth process affects the ontogenetic development of motor abilities in organisms, the actual mechanisms are far from being understood. One of the key open questions concerns the relationship of growth to the degrees of freedom coordination (DoF) problem [62]; one proposed coping mechanism is to freeze and progressively unfreeze some of the DoF in parallel to growth and motor skill acquisition [63]. An alternative strategy for redundancy resolution is control based on a reduced dimensional representation of a system’s behaviour. In this context, the theory of motor primitives (or muscle synergies) [60, 64], suggests a strategy enabling control dimensionality reduction through enforcement of linear constraints in the input. It has been proposed that the motor primitives may be obtained by computation using a reduced dimensional model of the musculo-skeletal dynamics [60]; the number of primitives required is simply the number of dimensions to which the system can be reduced to. However, dimensionality and therefore the reducibility of a mechanical system, is dependent on the physical parameters within the mechanical system.

In this work, the effect of some parameter variations, i.e. of mass and damping, on the reduced dimensionality is empirically studied in a simulated system consisting of a 1D chain of 10 masses interconnected with linear springs and damping elements. The chain is fixed in the proximal end, and free to move in the distal end, as depicted in Fig.5.2 and each mass element can be independently actuated by applying forces. The system is perturbed
and the resulting behaviour is used for dimensionality reduction using Proper Orthogonal Decomposition (POD); the change in dimensionality is analysed for various parameter values. Although a relatively simple model, we loosely compare it to vertebrate limbs and try to gain insight and generate testable hypothesis on the effect of physical growth on the dimensionality of control.

5.2.1 Experiments and Results

We investigate the effect of physical variations (growth of parameters) on reduced dimensionality for the following parameters:

1. Ratio of Damping to Stiffness (Damping Ratio) $D_{ri} = c_i / k_i$: quantifies the local stiffness along the chain,

2. Ratio of Mass to Stiffness (Mass Ratio) $M_{ri} = m_i / k_i$: quantifies the relative weight along the chain.

The stiffness is maintained constant throughout (at 1 N/m) and the parameter of interest $p_i$ at position $i$ ($M_{ri}$ or $D_{ri}$) is subjected to a change $\Delta p_i = f(i, g)$, where $g$ is an integer denoting the growth rate of the parameter. The growth rate could be thought of as a loose analogy to the effect of time on the natural growth process since growth seems to occur in quanta.$^{65}$

The experiments which were carried out on an increasing growth explored the following 2 Scenarios: a) Uniform Growth throughout an arm with equal initial distribution, $p_i = a + bg$; b) Uniform Growth throughout an distally distributed arm, i.e. $p_i = (a + bi)g$. The constants $a$ and $b$ are dependent on the range of variability of the parameter. Based upon these 2 scenarios, the experiments performed studied 2 kinds of growth effects, i) Damping ratio decrease; ii) Mass ratio increase.

For each run, the simulated system is perturbed by fixed duration pulse inputs. Then POD, (principal component analysis on the state trajectory, i.e. positions and velocities of the masses), is used to compute a reduced dimensional model (which can then be used for control), by truncating to the minimum set of normalised components below a threshold $t\%$.

For the first experiment, the damping ratio was decreased in 25 steps of 0.0196N/m/s starting from, 0.5N/m/s uniformly for scenario A, and a proximal to distal linear distribution in the range [0.125, 0.01]N/m/s for scenario B. For the second experiment, the mass ratio was increased in 25 steps of 1kg starting from, 0.5N/m/s uniformly for scenario A, and a proximal to distal linear distribution in the range [6.25, 1]kg for scenario B.
Figure 5.1: Dataset resulting from babbling : example from uniform parameter change (a) Damping change, (b) Mass change.

The results on dimensionality increase in Fig. 5.2 show a much greater dependency on damping ratio decrease (a), than on mass ratio increase (b). It is known damping tends to decrease dimensionality in mechanical systems, due to its effect of attenuating the higher frequencies in a system [66] the result in Fig 5.2a demonstrates a similar trend. On the other hand, while in principle, mass changes should not affect the dimensionality since it uniformly affects all the frequencies. However due to the effect of using a fixed time step for the POD analysis, a small increase in the dimensionality results, as can be seen in Fig 5.2b.

5.2.2 Discussion

Dimensionality reduction may be an essential component of the growth and development process of complex organisms and our framework allows us to
test this relationship. Although the increase in mass during the growth process is a necessity, the results leads us to hypothesise that from a dimensionality and learning perspective, it might be beneficial to maintain a large degree of damping towards distal ends of limbs and progressively decrease it along with increase in the mental abilities, in order to optimally aid control development.

5.3 Effect of Basis Functions on Reduced Dimensionality

Among the various challenges confronting the field of control of complex mechanical systems, the “curse of dimensionality” is well known; an exponential increment in complexity as a function of the state space. This renders known techniques of control such as optimal control, adaptive control etc. of many systems intractable. An approach that has been proposed to deal with this complexity is to seek a reduced dimensional representation of the mechanical system that can capture the input-output relationship to the best extent possible; such a representation can then be utilised to generate control signals for the original system. A well known reduction framework in this regard is known as projection [67].

There exists some biological evidence for the existence of control strategies that achieve some form of dimensionality reduction within our neural control mechanisms, leading to the hypothesis that this principle might underlie biological movement control. In particular of interests are the theories of muscle synergies [68] and the related notion of motor primitives [60] apart from models of biological pattern generation mechanisms [69]. A variety of theories and models have been proposed to explain the organisation of motor control at various levels in the structure (control mechanisms seem to be organised hierarchically). There is however a consensus on the notion of an architecture possessing modularity that linearises the control problem, and that employs stored pattern mechanisms.

In this context, an idea which has arisen in recent times is to actuate the system through a controller which comprises of a set of basis functions in time. The basis functions in this representation are chosen to be a platform and task independent controller formulation, such that the only thing that needs to be learned to suitably perform tasks are the weights by which they may be combined across the various actuators. For the purposes of this report, polynomial basis functions are assumed, where the basis functions are polynomial functions of time.
The rationale behind the research being reported on is that Effective Dimensionality can be used as a standard to measure behaviour of dynamical systems. In particular the idea of dynamical systems driven by pattern based controllers is often analysed from perspective of learning tasks. The tasks themselves are seldom compared in works dealing with pattern based controllers. The framework we present allows comparison of “tasks” when a dynamical system is driven by a bank of pattern based controllers. This is done by obtaining an equivalent dynamical system thus incorporating the weights that specified the control. The input in this case is a bank of pattern generating functions.

Dimensionality is a important but often overlooked in learning of motor control. A point mass is used for these experiments because it is a simple dynamical system which can be used to understand aspects of the problem. Good enough to test the framework presented. An analysis that can be expanded to a more general case if necessary. Polynomial basis functions are used because it is a simple basis to understand, easy to differentiate/integrate so can be exploited for derived closed form analytical solutions. Although numerical rather than analytic in flavour, the aims of this report are to present a novel framework by which task-specific dimensionality reduction may be quantified and studied. The scope of the control is only Feedforward control and not feedback.

The four results reported in this work:

- Minimum Dimensional Models : Parameterization, Can the models be parameterised for the task? Maybe.
- Numerical Optimisation of Dimensionality in performing tasks Finding the minimum dimensional trajectory satisfying task constraints (boundary value problem).
- Trajectory Comparisons : Set of Random trajectories: comparing the cost of computing control

5.3.1 Basis Function Control

Consider a dynamical system of the form,

\[ \dot{x} = f(x) + g(u), \quad y = h(x), \]  

(5.1)

where \( x \) is the state variable, \( u \) is denoted the input, and \( y \) as the output.

In its simplest formulation, a basis function controller can be represented by decomposing the input \( u(t) \) into a linear summation of basis functions represented by,
\[ u(t) = \Psi(t) = \sum_{i=1}^{n} w_i \psi_i(t), \quad (5.2) \]

where set of functions \( \Psi(t) = [\psi_1(t) \ldots \psi_n(t)] \) are the new basis in which the input \( u(t) \) is defined and the set \( W = [w_1 \ldots w_n] \) are the weights denoting a particular input. The basis functions are now functions in time, which remain common for all kinds of input signals passed onto the original system. The weights must however be computed carefully in order to achieve a desired trajectory. An alternative choice is to define the basis functions in the desired output (desired trajectory) and then computing the input through an “inversion” of the original system.

**Basis Functions for Inverse Control**

In this kind of a system, we specify that the aim of control is to achieve an desired state (or state sequence in time) in the output \( y_d(t) \). The basic premise of basis function control is that we decompose the desired output in the form as,

\[ y_d(t) = \sum_{i=1}^{n} w_i \psi_i(t), \quad (5.3) \]

where set of functions \([\psi_1(t) \ldots \psi_n(t)]\) are the new basis in which the output \( y(t) \) is defined and the set \( W = [w_1 \ldots w_n] \) are the weights denoting a particular desired trajectory \( y_d(t) \). We then seek to recompose the input \( u(t) \) using this set of weights. This is done by defining the input in terms of desired state, by inverting the dynamics.

In the linear system case, the system is defined as,

\[ \dot{x} = Ax + Bu, \quad y = Cx, \quad (5.4) \]

in which case, the inverse of the dynamics for a desired output \( y_d(t) \) is given by,

\[ u = B^{-1}[\dot{x}_d - Ax_d], \quad (5.5) \]

which can be rewritten in terms of \( y_d(t) \) as\(^1\),

\[ u = B^{-1}[C^{-1}\dot{x}_d - AC^{-1}x_d]. \quad (5.6) \]

Some types of basis functions that often encountered in literature are the Polynomial Basis \( (\psi_i(t) = t^i) \), and the Gaussian Basis \( (\psi_i(t) = \exp(k(t - \]

\(^1\)Note that in the case of \( B \) or \( C \) being not square, the usage of the pseudo inversion (moor-penrose inverse) \( B^\dagger \) and \( C^\dagger \) may be necessary, although not guaranteed to work.
Of interest to roboticists / biologists are the decompositions to more complicated functions of time, such as to sets of nonlinear dynamical systems, such as central pattern generators (CPG) \cite{69} or, dynamical systems themselves utilising basis functions, eg. Dynamic Movement Primitives (DMP) \cite{70}

**Polynomial Basis**

Polynomial Basis function $\Psi(t)$ are defined by using basis which are polynomial functions of time $t$, i.e.,

$$\Psi(t) = [\psi_1(t) \ldots \psi_n(t)]^T = [1, t, t^2, \ldots t^n]^T$$

The equivalent system under control is therefore,

$$\dot{x} = Ax + BW\Psi(t), \quad Y = Cx \quad (5.8)$$

### 5.3.2 Model Order Reduction

Often in physical systems, the behaviour is described by equations of motion derived from physical laws; these sometimes tend to be verbose in terms of number of equations. In most systems it is sufficient to understand the behaviour merely from the perspective of input and output. This is a requirement in domains such as control and simulation of dynamical systems. Model Order Reduction (MOR) deals with tools and techniques to produce a reduced dimensional (order) representation of a systems behaviour, which can then be employed in simulation, prediction and control.

Given a system which follows the following dynamics,

$$\dot{x} = f(x) + g(u), \quad y = h(x),$$

where $x \in \mathbb{R}^n$, an $n$ dimensional space, and the inputs $u \in \mathbb{R}^i$ and outputs $y \in \mathbb{R}^o$ are such that $i$ and $o$ need not be equal to $n$. In this case, the representation we are seeking aims to find an equivalent system,

$$\dot{z} = f'(z) + g'(u), \quad y = h'(z),$$

where $z \in \mathbb{R}^k$, where the new state variable $k \ll n$. Note that the inputs $u$ and outputs $y$ do not change.

The Projection Framework aims to find a reduced dimensional representation of the dynamics of a systems by “projecting” the state of the full
dimensional system into a lower dimensional subspace. The aim of the projection framework is to find a mapping $W$ in,

$$z = Wx,$$  \hspace{1cm} (5.9)

such that certain conditions are met in the input output relationship. The various methods approach this computation differently. For more information on popular model reduction techniques, the reader is referred to surveys such as in [67]. Balanced Truncation, which is the technique employed in this report is introduced in Appendix.

The question of how many states a system can be reduced to, is crucial for this work, since it is used as the metric to quantify one kind of controller over another (as well as one kind of task against another). For reduction methods based on projection, a convenient metric that can be employed is a threshold percentage $t_r\%$ to quantify the quality of the new reduced dimension.

If balanced truncation is employed, the Hankel Singular Value $[\sigma_1 \ldots \sigma_k]$ are a measure of the “control energy” for each of the new balanced states, the “energy” in this case referring to a product of the controllability and observability of a given state. Thus the threshold $t_r\%$ quantifies the percentage of control energy retained by the new reduced dimensional system, which is computed by normalising the hankel singular values and computing their relative percentages. Thus the number of states a given system can be reduced to depends on how concentrated the magnitudes of the normalised hankel singular values are.

### 5.3.3 Test System: Free Mass in 2D space

The system that was used for the analysis presented in this report is a kinematically (i) free moving mass in 2D Space as depicted in Fig. 5.4a. Variations of System include “tethering” the mass to a point in space using (ii) linear springs and (iii) applying damping forces on its motion as in Fig. 5.4b.

For all three kinds of systems, the dynamics are written in the form of Eq.5.4 as,

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{v} \\ \dot{w} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -k_x & 0 & -b_x & 0 \\ 0 & -k_y & 0 & -b_y \end{bmatrix} \begin{bmatrix} x \\ y \\ v \\ w \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} F_{ux} \\ F_{uy} \end{bmatrix},$$  \hspace{1cm} (5.10)

where $v = \dot{x}$, and $w = \dot{y}$. For the case of free mass, the springs and dampers $(k_x, k_y, b_x, b_y)$ are all fixed at 0, while for the second case, that of a springy mass, only the dampers $(b_x, b_y)$ are set to 0. The third case uses $+ve$ values of springs and dampers.
The first task was to analyse the stability of the three kinds of systems provided. The stability in each of these cases is analysed using the eigenvalues of the state matrix \( A \).

\[
\text{eig}(A) = \frac{1}{2} \begin{bmatrix}
-\sqrt{b_1^2 - 4k_1} \\
\sqrt{b_1^2 - 4k_1} - b_1 \\
-b_2 - \sqrt{b_2^2 - 4k_2} \\
\sqrt{b_2^2 - 4k_2} - b_2
\end{bmatrix},
\]

which gives the result of free mass \( (a) = [0, 0, 0, 0]^T \), springy mass \( (b) = [-i\sqrt{k_1}, i\sqrt{k_1}, -i\sqrt{k_2}, i\sqrt{k_2}] \). So system \( (b) \) is marginally stable since input impulses can persist infinitely. Since the stability is measured if the real component of the poles are negative, and the.

5.3.4 Reaching Problem

Reaching problems are control problems where we seek to compute the control input \( u \) that satisfies the following constraints on a trajectory \( x(t) \),

\[
x(t) = \begin{cases}
[x_0, y_0, \dot{x}_0, \dot{y}_0]^T, & t = t_0 \\
[x_f, y_f, \dot{x}_f, \dot{y}_f]^T, & t = t_f
\end{cases}
\]

(5.12)

Since the control is decomposed into the basis functions \( \Psi(t) \), each boundary value problem is solved by finding the appropriate set of weights \( \mathcal{W} \). This is done using a least-squares minimization.

First we compute the Vandermonde matrix \( \mathcal{V} \) composed of elements from \( t^i \) for fitting the boundary value problem,

\[
\mathcal{V} = \begin{bmatrix}
1 & t_0 & \ldots & t_0^{n-1} & t_0^n \\
t_0 & 2t_0^2 & \ldots & nt_0^{n-1} & 0 \\
t_f & 2t_f^2 & \ldots & nt_f^{n-1} & 0 \\
1 & t_f & \ldots & t_f^{n-1} & t_f^n
\end{bmatrix},
\]

(5.13)

then the required weights are obtained as,

\[
\mathcal{W}_x = \mathcal{V}^{-1} [x_0, \dot{x}_0, x_f, \dot{x}_f]^T
\]

(5.14)

\[
\mathcal{W}_y = \mathcal{V}^{-1} [x_0, \dot{x}_0, y_f, \dot{y}_f]^T
\]

(5.15)
5.3.5 Reaching with Way-point Constraints

These are special cases of the reaching control problem where a set of intermediate way-point constraints in position and velocity are specified. Here, we seek to compute the control input $u$ that satisfies the following constraints on a trajectory $x(t)$,

$$x(t) = \begin{bmatrix} x_0, y_0, \dot{x}_0, \dot{y}_0^T, & t = t_0 \\ \vdots \\ x_i, y_i, \dot{x}_i, \dot{y}_i^T, & t = t_i \\ \vdots \\ x_f, y_f, \dot{x}_f, \dot{y}_f^T, & t = t_f \end{bmatrix}$$  \hspace{1cm} (5.16)

Since the control is decomposed into the basis functions $\Psi(t)$, each boundary value problem is solved by finding the appropriate set of weights $\mathcal{W}$. This is done using a least-squares minimization.

First we compute the Vandermonde matrix $\mathcal{V}$ composed of elements from $t^i$ for fitting the boundary value problem,

$$c\mathcal{V} = \begin{bmatrix} 1 & t_0 & \ldots & t_0^{n-1} & t_0^n \\ t_0 & 2t_0^2 & \ldots & nt_0^{n-1} & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & t_i & \ldots & t_i^{n-1} & t_i^n \\ t_i & 2t_i^2 & \ldots & nt_i^{n-1} & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & t_f & \ldots & t_f^{n-1} & t_f^n \\ t_f & 2t_f^2 & \ldots & nt_f^{n-1} & 0 \end{bmatrix},$$  \hspace{1cm} (5.17)

then the required weights are obtained as,

$$\mathcal{W}_x = \mathcal{V}^{-1} \begin{bmatrix} x_0, \dot{x}_0, x_f, \dot{x}_f \end{bmatrix}^T$$  \hspace{1cm} (5.18)

$$\mathcal{W}_y = \mathcal{V}^{-1} \begin{bmatrix} x_0, \dot{x}_0, y_f, \dot{y}_f \end{bmatrix}^T$$  \hspace{1cm} (5.19)

With this formulation the learning of control for reaching is solved as a least squares optimization problem.

5.3.6 Experiments and Results

First 3\textsuperscript{rd} order polynomials are used and a least square fit is used to learn the weights to track a straight line trajectory. The first result is to compare a set
of minimum dimensional models. We present results in the form of variation in the model parameters (i.e. the input matrix $B_r$ and the output matrix $C_r$). The first result is the variation of the Reduced Order Model parameters with change in orientation of the target with respect to the origin. This is presented in Fig.5.5. The second result is the variation with distance of the target from the origin. This is presented in Fig.5.6. The key aspect of the these two results is that, the variation in the reduced order model parameters follow simple patterns which can be exploited in the form of a lookup table. This we can extrapolate/interpolate and generate appropriate reduced order models for novel target positions.

The effect of number of basis functions on the reduced dimensionality is presented in Fig.5.7. As it can be seen, the dimensionality of the controlled system is unaffected by increase in the number of basis functions used to represent the task. This is a key aspect of the usage of basis functions for control. They are used to decompose a desired trajectory and thus we simply increase the number of basis functions to accurately mimic a desired trajectory. However, the increase does not necessarily affect the dimensionality of the net system.

### 5.4 Numerical Optimisation of Dimensionality in performing tasks

Finding the minimum dimensional trajectory satisfying task constraints (boundary value problem). This is presented in Fig.5.8. An interesting feature was that optimal trajectories that result are straight line trajectories. This is a key result of this research showing that using polynomial time basis functions on a linear dynamical system results in minimum dimensional tasks lying on straight lines from the origin.
Figure 5.2: Test System and Results (a) A 1D chain of masses interconnected with springs and damping elements. Reduced Dimensionality for Parametric Variations: (b) Damping ratio decrease, (c) Mass ratio increase, under scenarios of even (Red) and uneven (Blue) parameter distributions.
Figure 5.3: Example of a reduction (a) State trajectories of system actuated by a set of basis functions combined by weights to produce a straight line motion, (b) Normalised Hankel Singular Values of the four states and the threshold

Figure 5.4: Free Mass in 2D Space (a) Unconstrained 2D motion, (b) Tethering with a spring and damper to the origin
Figure 5.5: Dimensionality and Variation of components of the reduced order model with orientation with respect to origin
Figure 5.6: Dimensionality and Variation of components of the reduced order model with distance with respect to origin

Figure 5.7: Dimensionality remains constant when the number of Basis Functions is increased.
Figure 5.8: Numerical Optimisation result (a) state trajectories of optimal result, (b) cartesian space trajectories of optimal result
Chapter 6

Muscle synergies in neuroscience and robotics: from input-space to task-space perspectives

6.1 Introduction

One of the fundamental questions in motor control concerns the mechanisms that underlie muscle contractions during the execution of movements. The complexity of the musculoskeletal apparatus as well as its dynamical properties allow biological systems to perform a wide variety of motor tasks [71]; on the other hand, such a complexity has to be mastered by efficient strategies implemented in the CNS. How does the CNS “choose” among the infinity of solutions of a given motor task (i.e. Bernstein problem) [62]? How are motor intentions translated into muscle activations? How can biological systems learn and plan movements so rapidly? A prominent hypothesis suggests that motor circuitries are organized in a modular fashion, so that muscle activations can be realized by flexibly combining such modules. Modularity has been observed in various forms such as kinematic strokes, spinal force fields and muscle synergies [59]; this paper provides an overview of the findings related to the so-called muscle synergies, as well as the application of such a concept in robotics and character animations.

Muscle synergies are defined as coordinated activations of a group of muscles. It has been suggested that the CNS encodes a set of synergies, and it

\[^{1}\text{The term synergy has also been used in the context of another motor control hypoth-}\]
combines them in a task-dependent fashion in order to generate the muscle contractions that lead to the desired movement (muscle synergy hypothesis). Evidence for this organization relies on the spatio-temporal regularities observed in the EMG activities of several species [72, 76]. Since in many cases these regularities appear to be very similar across subjects and motor tasks (i.e. robustness of muscle synergies), scientists have proposed that they might reflect a modular organization of the underlying neural circuitries. Assuming that muscle activations represent the control input to the musculoskeletal system, in this context muscle synergies are implicitly defined as input-space generators (i.e. components that are able to generate the necessary input signals).

From a computational point of view, a modular organization based on muscle synergies is very attractive. The activations of many muscles is hypothetically implemented by modulating the contributions of a small set of predefined muscle synergies. Such a dimensionality reduction may simplify motor control and learning, and it may contribute to the adaptability observed in biological systems [77]. This observation has recently motivated roboticists and control engineers to develop control strategies that are based on the same concept: combination of a small number of predefined actuations. In addition to the possible dimensionality reduction, the modularity of such scheme has the advantage that improved performance may be achieved incrementally by introducing additional synergies to the controller. The price to be paid is the restriction of the possible actuations to those that can be obtained by combining the synergies (i.e. synergies span set). This also implies a reduction of the possible movements that the controlled system can perform.

In the two fields of neuroscience and control engineering, research on muscle synergies is characterized by radically different goals and approaches (see Fig. 6.1). In the context of controlling artificial systems, the main goal is the synthesis of a small set of synergies that instantiates an effective control strategy. The obtained controller, as such, is mainly evaluated in relation to task-accomplishment, and in particular it should be able to generate a set of feasible actuations that allows the agent to perform a wide variety of tasks.

esis, the uncontrolled manifold hypothesis (UMH) [72, 73]. In that context, the term refers to “a neural organization of a set of elemental variables [e.g. muscle contractions] with the purpose to ensure a certain stability properties of a performance variable produced by the whole set [e.g. desired joint configuration]” [73]. These studies are out of the scope of this paper, however we will discuss the concept of M-modes, that has been introduced in the UMH but it is very similar to the definition of synergies we adopt in this manuscricipt.
In neuroscience, on the other hand, the main goal is to validate or falsify the hypothesis of muscle synergy. The typical approach consists in analyzing a dataset of recorded muscle activities, and in verifying if such a dataset is compatible with the proposed modular decomposition; the hypothetical synergies are inferred by applying a decomposition algorithm to the dataset of EMG signals. Unlike in control engineering, the major focus of this line of research resides at the motor level (i.e. the input-space of muscle activations); the evaluation of the hypothesized modular organization at the level of task is not always considered and, from our point of view, it deserves more attention. Does the set of muscle synergies identified from the EMG-dataset actually lead to the task performance observed experimentally? Does it generate feasible actuations? These issues have been investigated a-posteriori using realistic models of the musculoskeletal systems of different species [60, 78, 79]. Additionally, novel methodologies to deal with these challenges are starting to emerge in experimental neuroscience as well [80, 81]. We believe that a shift of paradigm from an input-space to a task-space identification of muscle synergies, which seems to be already in progress, may contribute to a better understanding of the hypothetical modularity of the CNS, and of its relationship to human learning and control. In particular in this review we argue that task-space constraints could be directly integrated in the decomposition algorithm used to extract the synergies.

This paper reviews the studies that investigate the hypothesis of muscle synergies, as well as the methods to control artificial systems that have been developed taking inspiration from this hypothesis. The organization of the paper follows the rationale developed so far. Initially, in Sec. [6.2] we provide a mathematical formulation of the concept of muscle synergies, we detail different synergy models (proposed as the mechanisms to generate muscle contractions), and we analyze their computational implications. In Sec. [6.3] we discuss the works that evaluate the hypothesis of muscle synergies solely in the space of the input-signals, and the ones that seek more direct neural evidence. Then, in Sec. [6.4] we present the studies that evaluate synergies also at the task-level; this section includes robotics, characters animation, as well as neuroscience. Finally, in Sec. [6.5] we offer further discussions and concluding remarks.

### 6.2 Models of muscle synergy

The concept of muscle synergy has been formalized in a variety of mathematical models. We will present these models in the context of controlling
Figure 6.1: Comparative scheme between research on muscle synergies in neuroscience and control engineering.

a generic dynamical system. This formulation is sufficiently generic to represent both the control of the musculoskeletal system and the control of an artificial agent. Furthermore, it is useful to explain the computational implications of the various synergy models, and to clarify the difference between input-space and task-space evaluation of a set of synergies.

The generic dynamical system we employ can be represented as follows:

$$\dot{x}(t) = f(x(t), t) + g(x(t), t)u(t),$$

where $t$ represents time, $x(t) \in \mathbb{R}^n$ is the system state variable at time $t$ (e.g. angular positions and velocities of the joints), and $u(t) \in \mathbb{R}^m$ is the system input at time $t$ (e.g. muscle activations or joint torques). Within this framework, the variable to be controlled is denoted as $y(t) \in \mathbb{R}^p$, and it is a generic function of the system state: $y(t) = h(x(t))$. The task is defined in terms of a set of constraints applied on the time evolution of this variable. Typical examples of tasks include reaching ($y(t_f) = y_d$ where $t_f$ is the desired reaching time), and tracking ($y(t) = y_d(t) \forall t$, where $y_d(\cdot)$ is the desired trajectory to be tracked). We refer to the task-space, as the space where the task $y_d$ is defined; similarly, the input-space is the space of the input signals $u(\cdot)$. The relation between these two spaces is given by the dynamics of the system. It is now clear that a given control input should always be evaluated in relation to the error between the corresponding evolution of the controlled variable and the desired task; in other words, it should always be evaluated in task space.
Classically, control inputs $\mathbf{u}(\cdot)$ belong to the infinite dimensional space of continuous functions. Under this assumption a number of interesting control properties (e.g. controllability and observability) can be proven. The idea behind modular control, is to significantly restrict the control input space by constraining $\mathbf{u}(\cdot)$ to be a combination of modules, or muscle synergies. The various muscle synergy models can be distinguished based on the mathematical formalization of this combination, and they are described in the following (see Fig. 6.2 for a schematic representation). An empirical comparison of these models when applied to a simple well-studied motor task is proposed by Chiovetto et al. [82].

**Temporal and synchronous synergies** In these models, the control input is defined as a linear combination of $k$ vectors $\mathbf{w} \in \mathbb{R}^m$, with 1-dimensional time-varying coefficients $a(t) : \mathbb{R}^+ \rightarrow \mathbb{R}$:

$$\mathbf{u}(t) = \sum_{j=1}^{k} a_j(t) \mathbf{w}_j.$$  \hspace{1cm} (6.1)

Each vector $\mathbf{w}_j$ specifies a balance between the input variables (e.g. balance between muscle activations), and its coefficient $a_j(t)$ determines its temporal evolution. In the temporal synergy model, the coefficients $\{a_j(t)\}$ serve as the task-independent predefined modules, and the vectors $\{\mathbf{w}_j\}$ represent the new (task-dependent) control input. As a result, this model reduces the control space to $k \cdot m$ dimensions; i.e. the $k$ m-dimensional vectors $\mathbf{w}_j$ have to be appropriately specified to fulfill the desired task $\mathbf{y}_d$. Synergies are then interpreted as the temporal patterns that are recruited selectively by different muscles. In literature, temporal synergies are also referred to as temporally fixed muscle synergies. An important special case, the premotor drive model, is obtained by defining the temporal coefficients as $a_j(t) = A_j \phi(t - \tau_j)$. In this case, the time course of the vectors $\mathbf{w}_j$ are determined by a common function $\phi(t)$, called premotor drive or burst pulse, that can be modulated in amplitude and shifted in time. In contrast, the synchronous synergy model defines the task-independent synergies as the vectors $\mathbf{w}_j$. The new control input $\{a_j(t)\}$ belongs to the infinite dimensional space of the one-dimensional real functions. Therefore this model, unlike the previous one, provides a dimensionality reduction only if the number of synergies is lower then the number of input variables, i.e. $k < m$. Synchronous synergies are co-varying group of muscles, and are also called time-invariant synergies, spatially fixed muscle synergies, or muscle modes.
Time-varying synergies  This model defines the control input as the superposition of \( k \) task-independent vector-valued functions \( w(t) : \mathbb{R}^+ \to \mathbb{R}^m \):

\[
u(t) = \sum_{j=1}^{k} a_j w_j(t - \tau_j).
\]

(6.2)

Each synergy \( w_j \) can be scaled in amplitude and shifted in time by means of the coefficients \( a_j, \tau_j \in \mathbb{R} \). These coefficients represent the new control input, and have to be chosen in order to accomplish the task \( y_d \). As a result, the new input space is reduced to a \( 2 \cdot k \) dimensional space. Neuroscientifically, these synergies are genuine spatiotemporal muscle patterns which do not make any explicit spatial and temporal separation. As such, according to this model, muscles within the same time-varying synergy do not necessarily co-vary.

6.3 Synergies as input-space generators

As discussed above, muscle synergies can be considered as input-space generators. Whether or not these generators are implemented in the CNS, and how they are eventually coordinated through the sensorimotor loops, is a main stream of research in motor neuroscience. To tackle this question, scientists have employed two main approaches. One of them is solely based on the analysis of EMG signals, therefore it can only provide indirect evidence of a modular neural organization. The other approach aims at locating the areas of the CNS where muscle synergies might be implemented, therefore providing a direct evidence. These methodologies as well as the obtained results are discussed in the following.

6.3.1 Indirect EMG-based evidence

The classical approach to evaluate the hypothesis of muscle synergies consists in searching spatio-temporal regularities (i.e. synergies) in a dataset of muscle activities (Fig. 6.3, continuous green arrows). Such a dataset is obtained by recording the EMG signals from a group of subjects/animals that are performing some prescribed motor tasks. As such, this methodology is mainly based on considerations grounded at the input level. The possibility to discriminate the various task instances from motor signals represents the only (a-posteriori) task-related verification of the identified synergies (see Fig. 6.1).

Linear dimensionality reduction algorithms are employed to identify a small set of components (i.e. synergies) that approximate the EMG dataset.
according to the chosen synergy model (see Sec. 6.2). The number of synergies to be extracted has to be specified a-priori by the experimenter, as it constitutes an input parameter of the decomposition algorithm. The choice of the decomposition algorithm to be used depends on the assumptions made on the nature of the hypothetical muscle synergies (e.g. non-negativity, orthogonality, statistical independence etc.) [83]. Principal component analysis (PCA) [84] looks for orthogonal synergies that account for as much of the variability in the data as possible. Similarly, Factor analysis (FA) [85] seeks the smallest set of synergies that can account for the common variance (correlation) of a set of muscles. Independent Component Analysis (ICA) [86] maximizes the statistical independence of the extracted components, thus it assumes that synergies represents independent information sources. Non-negative matrix factorization (NMF) [87] enforces the extracted synergies and their activation coefficients to be non-negative; this constraint reflects the non-negativity of neural and muscle activations (â€œpull-onlyâ€œ behavior). Additionally, NMF does not assume that the generators are statistically independent, thus it is more compatible with the observation that activations of multiple synergies are correlated [88]. Finally, the extraction of time-varying synergies is performed by an NMF-based algorithm developed ad-hoc that allows the components to be shifted in time [89].

To assess the quality of the extracted synergies, the so-called VAF (Variance Accounted For) metric is typically used (see Fig. 6.1). VAF quantifies the percentage of variability in the EMG dataset that is accounted for by the extracted synergies. High values of VAF indicate good reconstruction of the recorded EMGs, which lends credit to the extracted synergy set; low VAF values cast doubt on the extracted synergies, indicating that they do not explain a large part of the EMG variance. This metric is also used for determining the dimensionality of the synergy space. The criteria used for this purpose rely on the assumption that most of the EMG variability is attributable to task-dependent muscle activations, whereas a small portion is due to several sources of noise. Under this assumption, the number of synergies is defined either by the point where the VAF-graph (i.e. the curve that describes the trend of the VAF as function of the number of synergies, which increases monotonically) reaches a threshold level (e.g. 90%) [90], or by its flattening point, i.e. the point where a drastic decrease of slope is observed. Such an “elbow” is in fact interpreted as the point that separates “structured” and noise-dependent variability, and therefore it can be used to define the minimum number of synergies that capture the task-related features [91, 92]. Besides the VAF metric, other metrics (e.g. log-likelihood [91]) have been proposed to evaluate the effectiveness of extracted synergies (still in input
space); a thorough discussion of these metrics is beyond the scope of the present review. As depicted in Fig. 6.1, this indirect methodology is mainly restricted to the analysis of input-level data. A complementary metric based on single-trial task-decoding techniques has been proposed by Delis et al. [81].

A significant amount of experiments has been conducted in frogs, cats, primates as well as humans in order to test the validity of the above-mentioned synergy models, and by extension, of the muscle synergy hypothesis itself. A pioneering study showed that a small set of synchronous muscle synergies could generate a large number of reflexive motor patterns produced by cutaneous stimulations of the frog hindlimb [93]. This study also demonstrated that microstimulation of the spinal cord produced very similar muscle synergies to the ones generated by the freely moving animal. Qualitatively similar synergies were also found by intraspinal microstimulation [88]. The above analysis was then extended in order to identify spatiotemporal patterns of muscle activities (i.e., time-varying muscle synergies) [68]. Few time-varying synergies were shown to underlie the muscle patterns required to let the frog kick in different directions, and their recruitment was directly related to movement kinematics. These findings were further generalized to a wide variety of frog natural motor behaviors such as jumping, swimming, and walking; evidence for both synchronous and time-varying synergies was reported [94]. Additionally, this study revealed that some synergies are shared across motor behaviors, while others are behavior specific.

The synergy models described in Sec. 6.2 do not include sensory feedback, however the original experiments on animals involved sensory-triggered reflexive movements. In fact, only a few studies have systematically investigated the influence of sensory feedback in the muscle synergy organization. Cheung et al. [95] analyzed the EMG signals collected from the bullfrog during locomotor behaviors before and after having interrupted its sensory pathways (i.e. deafferentation). Their findings support the existence of centrally organized synchronous muscle synergies that are modulated by sensory inflow. Further support was provided by showing that an appropriate modulation of the synergy activations could explain immediate motor adjustments, and that these synergies were robust across different dynamic conditions [96]. A discussion on the role of sensory feedback is provided in Sec. 6.5.

A number of studies have examined the generalization of the above results to other species. In primates, Overduin et al. [97] found that three time-varying synergies described a large repertoire of grasping tasks. Shape and size of the grasped objects were shown to modulate the recruitment
strength as well as the timing of each synergy. In this way, this study validated that time-varying synergies account for salient task differences, and their activations can be tuned to adapt to novel behavioral contexts. Along the same lines, Brochier et al. [98] provided further support for such a robust and distinctive synergistic organization of primates’ muscle patterns during grasping. Analysis of single-trial EMG signals demonstrated that the time-varying activation of three synchronous synergies was reproducible across repetitions of the same grasping task and allowed unequivocal identification of the object grasped in each single trial. In cats, Ting’s group showed that muscle synergies could be mapped onto the control of task-level variables; such experiments will be detailed in Sec. 6.4.2.

The framework of muscle synergies has been successful also in characterizing the spatio-temporal organization of muscle contractions during human reaching tasks. Muscle patterns observed during movements in different directions [99] and speed [100] were accurately reconstructed by appropriate linear combinations of synergies, which appeared very similar across subjects. The synergies that were extracted from muscle activities during unloaded reaching (i.e. subjects did not hold any load in their hands) accounted for the EMG signals obtained during loaded conditions. The recruitment of the individual synergies, as well as their onset time, were consistently modulated with movement direction, and did not change substantially with movement speed. This observation was further confirmed by Muceli et al. [101]; in this study a small set of specialized synchronous synergies was able to explain a large set of multijoint movements in various directions. Finally, visually guided online corrections during center-out reaching were tested recently. The synergistic strategy was shown to be robust and more effective in explaining the corrective muscle patterns than the individual muscle activities [102]. Furthermore, it was shown that to correct the ongoing reaching movement, the CNS may either modulate existing synergies [102], or reprogram new ones [103].

Roh et al. [104] showed that an appropriate set of synergies could reconstruct the average patterns of muscle activation observed during isometric forces production in humans. The EMG signals were obtained for different force magnitude, directions and initial postures. The extracted synergies were very similar across conditions, and they were able to explain the corresponding datasets. Each synergy seemed to underly a specific force direction, while its activation coefficient appeared correlated to the force magnitude. In another series of experiments, a small set of synchronous synergies was able to explain static hand postures and discriminate the shapes of grasped
A relevant series of experiments showed that muscle activations involved in human postural control can be explained in terms of combinations of muscle synergies. A set of synchronous muscle synergies was able to explain muscle activations involved in postural stabilization; the EMG variation observed among trials and perturbation directions was accounted for by appropriate modulation of the synergies activation coefficients. In order to verify that the extracted synergies did not depend only on the specific biomechanical context, in a new experiment a set of subjects were asked to react to support perturbation from different postural configurations. The extracted synergies were very similar across the different conditions; however, in some cases task-specific muscle synergies needed to be added to the original synergy set to obtain a satisfactory EMG reconstruction. As the various postures lead to different patterns of sensory inflow, these results rule out the possibility that the observed synergies are only determined by specific patterns of sensory stimulations. On the contrary, they support the hypothesis that different muscle postural responses are generated by task-related modulations of the synergy activation levels. Such a hypothesis found evidence in the experiments performed by Safavynia and Ting, where the temporal recruitment of the synchronous muscle synergies were explained by a mathematical model that explicitly takes into account the kinematic of the subject’s center-of-mass (CoM). The authors then concluded that synchronous muscle synergies are recruited according to an estimate of task-related variables. The same model was previously used to fit the activations of each muscle independently during the same postural perturbation tasks. Related to postural control, Krishnamoorthy and colleagues analyzed the muscle activations that underly shifts of the centers of pressure (COP) of standing subjects. In this experiment three “muscle modes”, extracted by means of PCA, explained most of the variability of the integrated EMG signals. Such components are equivalent to synchronous muscle synergies as defined in Sec. 6.2 and they are characterized by the authors as the independent elemental variables that are controlled synergistically (in the sense of the UMH) by the CNS to stabilize the COP. Specifically, the model assumes that the location of the COP is modified by linear combinations of the M-modes, and their mixing coefficients represent the independent variables controlled by the CNS. Perreault et al. examined the organization of reflexes involved in postural stabilization in both stiff and compliant environments and showed that, although reflexes are altered between the two
environments and modulated according to the direction of perturbation, the synchronous muscle synergies underlying these changes are very similar.

Another scenario that provides evidence to the hypothesis of muscle synergies is human locomotion \cite{114,115}. Ivanenko et al. \cite{116} showed that five temporal synergies could reconstruct the muscle activity involved in locomotion tasks. These patterns are robust across walking speeds and gravitational loads, and they relate to foot kinematics \cite{114}. Additionally, the same temporal synergies (accompanied by additional ones) were observed during the coordination of locomotion with additional voluntary movements \cite{118}. Similar results have been reported in other locomotor behaviors such as running \cite{119} and pedaling \cite{120}.

Finally, some experiments have investigated how the hypothetical synergy organization of the CNS evolves during ontogenetic development \cite{121}. Dominici et al. \cite{122} observed that the two temporal synergies identified in stepping neonates are retained through development, and they are augmented by two new patterns first revealed in toddlers. The final set of synergies was observed in several animal species, consistent with the hypothesis that, despite substantial phylogenetic distances and morphological differences, locomotion is built starting from common temporal synergies. This conclusion was also supported by the comparison of temporal synergies extracted from young and elderly people, which revealed no significant effect of aging on synergy compositionality and activation \cite{123}.

### 6.3.2 Direct neural evidence

The studies presented so far support the existence of synergistic muscle activations during the sensorimotor control of movements. However, these methods are indirect, in the sense that the presence of synergistic structures within the CNS can only be inferred. What remains to be tested is whether the uncovered muscle organization is neurally implemented in the CNS and, if so, in which areas. Alternatively, one could argue that the extracted synergies represent a phenomenological output of the motor coordination required for movement execution. For instance, recently Kutch and Valero-Cuevas \cite{124} designed carefully thought experiments and simulations to show that muscle synergies can be observed even if the nervous system does not control muscles in groups. The authors demonstrated that muscle synergies, as detected via dimensionality reduction methods (see Sec. 6.3.1), may originate from biomechanical couplings and/or from constraints of the task. Similar conclusions were already reached by Valero-Cuevas et al. \cite{125}, who showed that
the within-trial variability of EMG data observed during the production of a vertical fingertip force was incompatible with the (unique) associated muscle synergy that would have been extracted. Although these findings do not directly falsify the muscle synergy hypothesis, they cast at least some doubts about the sole neural origin of modularity.

This underlines the need for a more critical assessment of the validity of the muscle synergy hypothesis. In this direction, a number of recent studies sought evidence for a neural implementation of muscle synergies, and examined which regions of the CNS may express synergies and their activations. This question has been addressed by attempting to relate neural activity with simultaneously recorded muscle activity during performance of different motor tasks. Using such an approach, Holdefer and Miller [126] provided direct support for the existence of neural substrates of muscle synergies in monkey’s primary motor cortex. In particular, they studied the activity of neurons and muscles during the execution of a variety of reaching and pointing movements, and they found that the discharge of individual neurons represents the activation of functional groups of muscles. In addition, Hart and Giszter [127] showed that some interneurons of the frog spinal cord were better correlated with temporal synergies than with individual muscles. Therefore, they suggested that these neural populations constitute a neural basis for synergistic muscle activations [128]. Another study demonstrated that the sequential activation of populations of neurons in the cat’s motor cortex initiates and sequentially modifies the activity of a small number of functionally distinct groups of synergistic muscles [129]. Similarly, Overduin et al. [130] showed that microstimulations of specific regions of the motor cortex of two rhesus macaques corresponded to well-defined spatial patterns of muscle activations. These synchronous synergies were very similar to those extracted from the same animals during natural reaching and grasping behaviors. Extending this research line in the context of motor learning, Kargo and Nitz [131] showed that early skill learning is expressed through selection and tuning of primary motor cortex firing rates, which specify temporal patterns of synergistic muscle contractions in the frog’s limb. Finally, Roh et al. [132] analyzed the muscle patterns of the frog before and after transection at different levels of the neuraxis: brain stem, medulla and spinal cord respectively. They found that medulla and spinal cord are sufficient for the expression of most (but not all) muscle synergies, which are likely activated by descending commands from supraspinal areas. Similarly, Hart and Giszter [133] examined the compositionality of temporal synergies in decerebrated and spinalized frogs. Their results indicated that in both cases temporal synergies consisted of pulsed or burst-like activations of groups of
muscles. They also showed that brainstem frogs had more focused muscle groups and showed richer behavior than spinalized equivalents.

In humans, the main approach to locate hypothetical muscle synergies has been to analyze brain-damaged patients. Comparing the synergies extracted from healthy and brain-damaged subjects could provide hints about the neural centers involved in the synergistic control of muscles. In this vein, examining motor tasks involving arm and hand movements, Cheung et al. [134] showed that the synchronous synergies extracted from the arm affected by a stroke were strikingly similar to the ones extracted from the unaffected arm, concluding that muscle synergies were located in regions of the CNS that were not damaged. In a second study involving subjects with more severe motor impairment [135], they found that synchronous synergies may be modified according to three distinct patterns — including preservation, merging, and fractionation of muscle synergies — reflecting the multiple neural responses that occur after cortical damage. These patterns varied as a function of both the severity of functional impairment and the temporal distance from stroke onset. Similarly, Roh et al. [136] found systematic alterations of the upper limb synergies involved in isometric force production in stroke patients with severe motor impairment. However, these alterations did not involve merging or fractionation of normal synergies. Clark et al. [137] investigated the modular organization of locomotion in stroke patients. They found a coordination pattern consisting of fewer synchronous synergies than for the healthy subjects. These synergies resulted from merging of the synergies observed in healthy subjects, suggesting reduced independence of neural control signals. In contrast, Gizzi et al. [138] demonstrated that the temporal waveforms of the synergy activation signals, but not the synchronous synergies, were preserved after stroke.

Finally, a different but worth-mentioning approach has been the attempt to map the activity of leg muscles onto the alpha-motoneuron pools along the rostrocaudal axis of the spinal cord during human locomotion [139, 140]. Using this procedure, the authors could infer the temporal and spatial spinal motor output for all the muscles of the legs during a variety of human walking conditions, and relate them to the control of task-relevant variables such as center of mass displacements. Overall, their findings support the existence of some spinal circuitry that implement temporal synergies. The strength of this approach resides in the explicit use of anatomical and clinical charts that document the innervation of the lower limb muscles from the lumbosacral enlargement [141].
6.4 Synergies from the perspective of the task-space

6.4.1 From input-space to task-space: general rationale

The methodology presented in Sec. 6.3.1 undeniably led to many crucial insights, however it does not guarantee that the extracted synergies account for the observed task performance. VAF-like metrics only measure the capability of the synergies to reconstruct/fit the dataset of recorded “input-signals” (i.e. EMG data). Moreover, in some studies, such signals are averaged across movement repetitions. In this case, the VAF constitutes an average indicator, and it does not quantify the capability of the synergies to reconstruct each individual trial [142]. Since the musculoskeletal apparatus is a non-linear system, these approximations of the recorded muscle activities may not lead to the observed task performance ([143], paragraph 1.1), a condition that would harm the validity of the hypothesized modular control structure. On a similar note, the extracted synergies might generate unfeasible joint torques. Finally, even if the dataset of muscle activity is very well approximated, additional muscles that are not recorded during the experiment might have a crucial role in the generation of the movement. These issues emerge because the dynamics of the musculoskeletal system (i.e. its input-output relation) is not directly taken into account in the synergy decomposition algorithms.

In this section we review those works that attempt to relate muscle synergies to performance variables defined in task-space. Initially, we present the concepts of functional synergies and spinal force fields. The former constitutes a valid strategy to include the task variables into the classical EMG-based analysis; the latter provides task-based evidence for neurally implemented muscle synergies. Then, we discuss some studies that, in the context of biomechanics, employ plausible musculoskeletal models to test the movements obtained from experimentally extracted muscle synergies. Finally, we shift our attention to robotics and characters animation. In these fields, the main challenge is the synthesis of a small set of synergies that reduces the dimensionality of control and, at the same time, spans a subspace of actuations that allows the agent to perform a wide variety of tasks (Fig. 6.3, red arrows). Ideally, the synthesized synergies should preserve controllability and reachability of the system. Loosely speaking, this means that any desired system state can be reached by an appropriate control input (i.e. combination of
synergies) in a finite amount of time. At the motor level, it is important that the synergies generate feasible actuations; additional properties, such as the generation of optimal control signals, may also be desirable (see Fig. 6.1).

6.4.2 Functional muscle synergies and spinal force fields

In most of the works presented so far, the functional role of muscle synergies is estimated a-posteriori by analyzing the dependence of the recruitment coefficients (i.e. gain and/or onset time) on the task conditions (e.g. reaching direction, force magnitude and direction, perturbation direction). Typically, each muscle synergy is assumed to underlie the task-level functionality observed in conjunction with the higher values of its activation coefficient. As an example, the analysis of directional tuning curves illustrated that some of the synergies were directly related to reaching in specific directions [100]. A different approach is taken by a pool of studies which define the concept of functional synergies; i.e. components, typically extracted by means of NMF, of a dataset containing both EMG signals and measurements of defined task-related variables. As a result, each component consists of two elements: a balance of muscle contractions (i.e. synchronous muscle synergy), and the evolution of the task-related variables induced by such a muscle synergy (task-related vector). In our view, the concept of functional synergies provides a way to tackle the drawbacks of input-based extraction algorithms: if a set of functional muscle synergies extracted from a training-set is able to reconstruct both the EMG and, more importantly, the task-related signals observed in another set of data (testing set), then it is more likely that combinations of such muscle synergies will generate the appropriate control signals to perform the task successfully.

Functional muscle synergies were analyzed in the context of postural tasks in experiments with humans [80] and cats [90, 144]. The task-related variables were defined as the forces measured under the feet of the subject, which reacted to unexpected motions of the support surface. The experiments showed that each subject exhibited the same functional synergies for both stepping and non-stepping responses to perturbations [80], suggesting that a common pool of muscle synergies, with specific biomechanical functionalities, can be used by the CNS to drive the motion of the CoM independently of the subject’s behavioral response. The functional synergies extracted from the non-stepping data were able to reconstruct the EMG signals, the CoM acceleration and the direction (not the magnitude) of the forces recorded during stepping responses; however, an additional stepping-specific muscle synergy was needed to increase the quality of EMG reconstruction. Generality and robustness of functional synergies were also analyzed in postural...
experiments with cats \cite{90}. In this study, a group of cats experienced both translations and rotations of the support surface. Functional muscle synergies were extracted from a dataset containing EMG signals and ground forces observed for different postural configurations (i.e. distances between the anterior and the posterior legs). The functional synergies extracted during surface translations for the most natural posture were able to reconstruct the data observed in all the other conditions (i.e. different postural configurations and surface rotations). Moreover, functional synergies appeared very similar across subjects. These results suggested that each muscle synergies implements a specific biomechanical functionality \cite{144}, which is general across tasks and robust across subjects.

The methodology proposed by Ting and colleagues is undoubtedly a valuable attempt to identify muscle synergies that are directly related to task execution, however it presents some limitations. First, NMF extracts non-negative components and coefficients; while this constraint is well justified at the muscle activation level (see Sec. 6.3.1), task variables may exhibit negative values. Second and more important, in addition to a linear superposition also at the task-level, it assumes that both EMG signals and task-variables are generated with the same mixing coefficients. Although it is possible to obtain a good fit of a given dataset, due to the non-linearity of the musculoskeletal system, this assumption does not hold in general.

A radically different approach to investigate the modularity of motor circuitries consists in analyzing the so called spinal force fields. This approach is grounded on the seminal discovery that electrical stimulations of individual regions of the frog’s spinal cord produce peculiar isometric endpoint forces that depend on the posture of the limb; the direction of the force vectors within each of these fields is invariant over time, while their magnitudes are characterized by a specific time evolution. Additionally, each of these force fields features a specific point of convergence. Structures with these characteristics can be generated by groups of coactive and linearly covarying muscles \cite{145, 146}. In particular, only a small subset of all the possible muscle combinations leads to robust and convergent force fields \cite{147}. Therefore, the observation of such characteristics in an experimentally measured force field can be regarded as an indirect evidence for spinally implemented temporal muscle synergies (see Sec. 6.2). Kargo and Giszter \cite{148} showed that rapid corrections of movements in wiping frogs can be explained as linear combinations of spinal force fields. Additional evidence was obtained by examining the force fields generated by frogs \cite{149} and turtles \cite{150} that exhibited deletion of motor patterns. Another method to investigate the nature of spinal
circuits is the analysis of feedback mechanisms in relation to force fields. Different external excitations of the frog’s muscle spindles during wiping reflexes led to structurally invariant force fields across time. Furthermore, the bursts of muscle activity underlying the wiping behavior and the balance of activations across muscles were not altered by the spindle feedback. Instead, feedback regulated the amplitude and the timing of each single burst. Since these variables did not covary across the pulses, the authors concluded that individual premotor drive pulses and not time-varying synergies are the units of spinal activity [151]. Such hypothetical neural organization is compatible to the synergy scheme proposed by Drew et al. [152] and Krouchev et al. [153] for locomotive behaviors. These schemes allow a sequential activation of coordinated groups of muscles, a mechanism that can be implemented in the premotor drive model by modulating the on-set time of the bursts. Spinal force fields are effectively task-level representations of hypothetical neural modules, however this methodology does not provide any estimate of what the corresponding muscle synergies may look like. Moreover, the relation between linear combinations of muscle synergies and linear combinations of force fields is far from being trivial.

6.4.3 Neuromechanical modeling

Although many studies in experimental motor control provide support to the hypothesis of muscle synergies, it is hard to test whether the proposed control model can effectively lead to the task performance observed experimentally and generalize to other tasks. This issue can be tackled computationally by employing biologically plausible models of the musculoskeletal apparatus.

A pool of studies investigate if a modular organization like the synchronous synergy model can explain a complex task like human walking [79, 154, 155]. A set of synergies are identified from a dataset of recorded EMG signals by means of NMF. Such “modules” are then used to generate the muscle control inputs to a musculoskeletal model of the human legs. Using these synergies as a first guess, a numerical procedure optimizes the relative level of muscle activation within each module and the time course of the weighting coefficients; the objective is to minimize the difference between the results of the forward simulation and the values of the task variables measured experimentally. The walking kinematic and the ground reaction forces are well reproduced by 5 modules, if the motion is constrained in 2D [79], and 6 modules for 3D walking [155]. Additional simulations reveal that the muscle groups identified during normal walking are able to emulate walking tasks with very different mechanical demands (i.e. change in mass and weight.
of the models). These results agree with the theoretical considerations formulated by Nori et al. Finally, this research shows that each module is associated to a specific biomechanical functionality (e.g. body support, forward propulsion, leg swing and balancing).

Related results are presented by McKay and Ting. In these studies the goal of the authors is to predict the patterns of muscle activities and the ground reaction forces observed experimentally in unrestrained balance tasks with cats. Muscle contractions of an anatomically-realistic musculoskeletal model of the cat are computed by an optimization procedure, that constrains task-related variables (i.e. center of mass) according to the experimental results. Although many different cost functions are tested, the best predictions are achieved by minimizing control effort (i.e. total squared muscle activation). Predictions improved if muscle contractions are constrained to linear combinations of the experimentally derived synergies; however the overall control effort increases, and the range of admissible ground forces reduces substantially. Furthermore, these studies validate the assumption made by Torres-Oviedo et al. that the ground reaction forces associated to each synergy rotate as a function of the limb axis. These results suggest that muscle synergies are feasible physiological mechanisms for the implementation of near-optimal or “good-enough” motor behaviors.

Kargo et al. employed a biomechanical model of the frog hindlimb to test whether the model of premotor drive could account for the wiping behavior observed experimentally. The parameters of the premotor drive model (i.e. muscle groups, pulse time course, and amplitude and phasing of the single synergies) are initially identified to reproduce isometric forces and free limb movement kinematics from a test limb posture. As expected, starting from different limb postures the derived feedforward control fail in driving the simulated limb toward the target. However, as showed by Kargo and Giszter, appropriate feedback modulations of the amplitude and the phase shift of the drive burst, and the adjustment of muscle balance based on the initial configuration of the limb, are enough to generate successful muscle activations. Furthermore, the limb trajectories obtained with and without feedback are very similar to those observed in intact and deafferented frogs respectively. These results support the model of premotor drives, in which feedback mechanisms preserve the duration of the pulses.

Berniker analyzed mathematically the control scheme of muscle synergies and proposed a principle for its formation. A linear reduced-dimensional dynamical model that preserves (to the best extent possible)
the natural dynamic of the original system is initially computed. Synergies are defined as the minimal set of input vectors that influence the output of the reduced-order model [161], and that minimally restrict the commands (and the resulting responses) useful to solve the desired tasks [60]. Practically, this set is found by optimizing the synergy matrix over a representative dataset of desired sensory-motor signals. This method was able to synthesize a set of synergies for the model of the frog hindlimb that were very similar to the ones observed experimentally [95]. Furthermore, the synergy based controller produced muscle activations and kinematic trajectories that were comparable with the ones obtained with the best-case controller (that can activate each muscle independently).

6.4.4 Robotics and character animation
In the context of robotics and characters animation, the concept of muscle synergies is appealing as it provides a strategy to reduce the number of variables to be controlled (synchronous synergy model), or more generically, the dimensionality of the control signals (time-varying synergy model). Animated characters are embedded in physical environments (i.e. dominated by physics laws) thus the associated control problem is totally equivalent to the control of a musculoskeletal model or of a humanoid robot. In this section we present the works that have been carried out in these fields of research.

The work proposed by Mussa-Ivaldi [162] is one of the first attempts to develop a controller based on the modularity observed in biological systems [163]. The idea is that the motion of a kinematic chain can be determined by a force field applied to its end effector. Inspired by the experiments performed by Giszter et al. [145], such a force-field results from the linear combination of basic fields, each characterized by a single equilibrium point in the operational space. The results obtained on a simulated two-degrees-of-freedom (dof) planar kinematic chain show that an appropriate choice of the basis-field coefficients can produce a wide variety of end-effector trajectories. Similarly, Matarić et al. [164] used force fields to drive joint torque controllers on a rigid-body animated character [165, 166]. Although the concept of spinal-force field is very similar, Mussa-Ivaldi’s work does not directly use the notion of synergy as defined in Sec. 6.2. A step forward is taken by Nori and Frezza [167], Nori [168], who propose a mathematical formulation for a set of actuations (i.e. synergies) that comply with the hypothetical properties of spinal-force fields [77]. The mathe-
vical description of the synergies is derived from the closed-form solution of an optimal control problem. Additionally, a feedback controller assures that the system follows the desired trajectory towards the synergy equilibrium position. It is proved that the proposed formulation guarantees system controllability\(^2\). The synthesized synergies are successfully tested on a simulated two-dof planar kinematic chain.

The idea that each synergy solves a well-defined control problem (e.g. to lead the system to the synergy-specific equilibrium position [167]), appears in several other studies [169–171]. Chhabra and Jacobs [170] propose a method called Greedy Additive Regression (GAR). A library of task-specific actuations (synergies) are kept in memory. When a new task has to be performed, a suitable actuation is initially searched in the linear span of these synergies. If the lowest task-error is above a certain threshold, the task will be solved via traditional methods (e.g. feedback error learning), and the obtained actuation will be added to the library. If the library already contains the maximum number of synergies allowed, the least used one will be removed. The obtained results suggest that synergies synthesized via GAR outperform primitives based on PCA if the dynamical system is non-linear (planar kinematic chain), and there is no statistical difference for linear systems. However, no theoretical explanation is provided.

In the same vein, Todorov [171] proved that, for a certain class of stochastic optimal control problems, an appropriate change of variable in the Bellman equation allows to obtain the optimal control policy as a linear combination of some primitives. These primitives are, in turns, optimal solutions of other optimal control problems. Such a method has recently been tested in the context of character animation [172]. It is important to clarify that this theory provides a theoretical grounding to the compositionality of optimal control laws, but like GAR it does not provide a method to compute such primitives. In fact, although new efficient methods have been proposed recently, solving an optimal control problem remains quite computationally intense, and it might be unfeasible for systems with a large number of dof.

Another mathematical framework, that has recently been developed in the context of character animations, is based on the optimal anechoic mixture decomposition model, mathematically equivalent to the time-varying synergy decomposition. Specifically, complex kinematic animations are obtained by

\[^2\text{In control theory, a system is said to be controllable if an external input can move the system from any initial state to any final state in a finite time interval.}\]
mixing primitive source signals that are learned from motion captured data\cite{173, 176}. Within this framework a number of interesting results have been achieved, including a mathematical proof of stability properties for groups of characters that interact in various ways \cite{177}.

The procedure presented by Alessandro and Nori \cite{169} is grounded on a method to solve generalized reaching tasks called Dynamic Response Decomposition (DRD). In this context, a task is defined as a set of constraints on the values of the state variables at given points of time. Initially a state-space trajectory that fulfills these constraints is computed by interpolating the task with a set of interpolants; finally, inverse dynamics is used to obtain the corresponding actuations. Based on this strategy, the following procedure allows to synthesize a set of synergies. The evolutions of the state variables (i.e. dynamic responses) to an extensive set of generic actuations (exploration phase) are used as the interpolants; in a second stage (reduction phase), they are used to interpolate a small set of tasks. The corresponding actuations proved to be effective synergies for additional reaching tasks on a simulated planar kinematic chain. Like the GAR method, this procedure generates synergies in the form of feedforward controllers, and it allows to modify incrementally the library of synergies. However, DRD provides a computationally fast method to solve the task. This technique has proved its efficacy empirically, but a solid theoretical grounding is still lacking.

Most of the methods presented so far require an accurate analytical model of the system dynamics. Such a model is not always available, and for certain robots, it might be difficult to identify. Todorov and Ghahramani \cite{178} propose a method to synthesize synergies by means of unsupervised learning. Their work emphasizes the role of muscle synergies in an hypothetical hierarchical control scheme similar to the one proposed by Safavynia and Ting \cite{109}: receptive fields translate sensory signals to internal variables, and muscle synergies translate high-level control signals applied to these variables to actual muscle contractions. From this perspective, receptive fields along with motor primitives must form an inverse model of the sensory-motor system. This mapping is learned by fitting a probabilistic model to a dataset of sensory-motor signals generated by actuating the robot with random pulses. The use of the learned synergies as low-level controllers substantially reduces the time needed to learn a desired policy, however their capability to generalize to additional control laws is not explicitly tested.

Alessandro et al. \cite{179} define synergies as parametrized functions of time that serve as feedforward controllers. The identification procedure consists
in finding the values of the parameters such that appropriate linear combinations of the resulting synergies drive the dynamical system over a set of desired trajectories (training set). The synergies identified are then tested for generalization; the idea is to evaluate to which extent they can generate actuations that drive the system along a new group of trajectories (testing set). This procedure has been evaluated successfully in simulation and does not require the analytical form of the system dynamics. However, it is computationally very intense as it involves heavy optimizations. In essence, this work proposes a new formal definition of the concept of muscle synergies: elementary controls that are evaluated in terms of task-performance (i.e. tracking error), rather than in terms of approximation of the input space.

Thomas and Barto [180] formulate the problem of primitive (i.e. synergy) discovery within the framework of reinforcement learning. In this case, the problem that the agent has to solve is a Markov Decision Process (MDP), and each primitive is a parametrized feedback control policy. The idea is to identify the optimal parameters that maximize the expected reward for a given task, when the control is restricted to linear combinations of the learned primitives. This method is tested on a simulated planar kinematic chain actuated with artificial muscles. Primitives are identified on reaching tasks, and they are successfully tested in another reaching task with the additional constraint of avoiding an obstacle. This work clearly shows the advantage of a synergy-based framework in terms of learning speed of novel control policies. This method is in essence similar to the one proposed by Alessandro et al. [179], however it identifies complete feedback control policies rather than single feedforward synergies.

The time-varying synergy model greatly reduces the dimensionality of the problems by encoding actuations with synergy-coefficients, however at the same time it introduces a complication. As the new input variables are piecewise constant, it is difficult (although possible) to implement feedback loops. The synchronous model ameliorates this problem and, to some extent, it allows adapting traditional control strategies to the new reduced-dimensional control input.

Some researchers employ the synchronous synergy model to control the tendon-driven robotic ACT hand [181] in a reduced dimensional space [182–184]. Similarly to Todorov and Ghahramani [178], dimensionality reduction is applied both in the sensory space and in the actuation space. The “observation synergies” transform sensory readings (tendon lengths) into a lower dimensional variable; the “control synergies” translates synergy-coefficients
(as defined in Sec. 6.2) to motor signals. Model adaptive control and PIDs are applied to the reduced dimensional input, and allow the robotic hand to perform tasks like writing\cite{183, 184} and playing piano\cite{182}. The synergy matrices (observation and control) are computed by applying PCA and NMF to a dataset of tendon-lengths obtained as a result of defined hand motions. It is noteworthy that the more this motions are similar to the ones required to solve the task, the better the quality of the obtained synergy-based controller. This is clearly not surprising, but it highlights the importance of task-related variables in the formation of muscle synergies\cite{185}.

Marques et al.\cite{186} identify synchronous synergies by means of an unsupervised Hebbian-like algorithm that captures the correlations between motor signals and sensory readings. Each synergy then summarizes the levels of correlation between each motor and one of the sensors. The time modulation of each synergy to solve a given task is then obtained by means of a supervised learning procedure that aims at reducing the task error. Unlike many other works in robotics, the exploratory strategy proposed to generate the dataset of sensory-motor data does not provide any information about the desired motor tasks, therefore muscle synergies are implicitly interpreted as patterns of motor coordinations that solely reflect the biomechanical constraints of the robot. This method has been tested on a single-joint tendon driven robot.

In the context of robotic hands, many researchers adopted the idea of postural-synergies, or eigengrasps. This concept derived by the observation that the variability of finger postures during human grasps can be explained by a few principal components\cite{187}, i.e. eigengrasps. Similarly, constraining the finger-joints positions of a robotic hand in such a way that the useful grasping postures can be obtained by superposing a small number of components, would result in a substantial simplification of the grasping problem. Ciocarlie and Allen\cite{188} derived a theoretical formulation of the problem of stable grasping in the low dimensional space of the postural-synergies; such a formulation is further improved by Gabiccini et al.\cite{189} for complain grasps. These studies are further analyzed and discussed by Bicchi et al.\cite{190}, who presented them from the point of view of modeling the process of grasping and active touch. Finally, Brown and Asada\cite{191} proposed a direct mechanical implementation of the eigengrasps. In all these works, the quantitative details of the postural-synergies are taken from human experiments and adapted to the robot mechanical structure; the problem of finding a set of synergies that is optimized for a given robotic hand is left as future research.
Reduced dimensionality based on postural synergies is also adopted by Hauser et al. [192] for the task of balancing a humanoid robot. The authors propose a mathematical formulation, as well as a method to construct kinematic synergies (i.e. predefined balance between joint positions) that are directly linked to task variables (e.g. for balance control, the center of pressure). Additionally, the synergies are constructed in such a way that the mapping from synergy coefficients to task variables is linear (similar to the work proposed by Nori and Frezza [167] but in kinematic space). This allows to use a simple PID on the synergy coefficients to control the center of pressure of the robot, as long as the movements are slow enough to neglect dynamic disturbances. The proposed method is demonstrated both in simulation and in a real humanoid.

As a final note, it is important to say that the concept of modularity has been employed in robot control in many other ways. In most of these works modules are defined as kinematic-based controllers that are combined sequentially to obtain complex joint trajectories [193, 194]. In this regard, these works are more related to the concept of kinematic stroke than to muscle synergies [195]. These works are out of the scope of this paper, as we focus on controllers that, in accordance with the models of muscle synergies, are based on (parallel) superpositions of primitives in input space.

6.5 Conclusions and perspectives

The hypothesis of muscle synergies, that proposes a modular organization of the neural circuitry involved in muscle coordination, has been proved very difficult to validate or falsify [196]. As discussed in Sec. 6.3, a substantial body of evidence in favor of this hypothesis comes from the observation that the main components of EMG recordings are robust across behaviors, biomechanical contexts, and individuals. In addition, the successful control of artificial agents confirm the computational feasibility of the hypothesized synergy-based controller (Sec. 6.4). However, there also exist experiments with the human hand that seem to disprove the hypothesis of muscle synergies [125, 197]. As a matter of fact, there is no real consensus yet on whether muscle synergies effectively represent a modular organization of the CNS, or they merely result from the methodology employed during the experiments.

The works that are based on the control of artificial agents (e.g. musculoskeletal models, robots, animated characters) clarify the importance of
evaluating synergies in task-space. In this context, the idea is to synthesize a set of synergies that guarantees the accomplishment of the desired tasks (Fig. 6.3 red arrows). On the contrary, the main focus of experimental motor control has been to identify the synergies that better reconstruct the recorded EMG dataset (Fig. 6.3 continuous green arrows), and to understand their neural substrate. This approach implicitly assumes that a well reconstructed input signal leads to the observed task performance. Given the non-linear dynamics of the musculoskeletal system, this assumption might not hold. For this reason, in our view the hypothesis of muscle synergies should be tested by validating an input-output model (i.e. from muscle activations to task-variables), rather than fitting a model of the input data alone (Fig. 6.3 dashed green arrow). In fact, we could speculate that muscle synergies encode a form of body schema [198] that allows translating intentions to motor plans (i.e. the inverse dynamic model of the musculoskeletal system) [108].

The concept of functional synergies represents a first attempt to relate muscle synergies to task variables. However, as discussed in Sec. 6.4.2, EMG and task-level components are assumed to be activated by the same coefficients. This assumption cannot hold in general because the musculoskeletal system is non-linear; rather, input-space and task-space coefficients should be related by a non-linear mapping (as described by Alessandro et al. [179]). To address this issue, one should go beyond the use of NMF, and develop novel techniques that do not impose a linear mapping between the two sets of coefficients. Additionally one could try to reconstruct the task-variables with more general non-linear methods instead of imposing a linear combination also at the task level. In the same spirit of the procedure used so far, such a technique should optimize the reconstruction error of the EMG signals, and constrain a good fit of the task-variables. In any case, the generality of the extracted functional synergies should be tested. To the best of our knowledge, the model of functional synergies was never used as a predictive framework. It would be extremely interesting to evaluate the extent to which functional synergies identified during the execution of a certain set of tasks, are able to predict the muscle activations observed during the execution of another task that involve the same task variables. If such prediction was unsuccessful, the experimenter could conclude that the identified muscle synergies do not really encode the hypothesized biomechanical functionalities, or that the same functionalities might be encoded by different synergies. In general, the model of muscle synergies has very seldom been used to make predictions.

An alternative strategy to verify the relationship between muscle synergies and task execution (Fig. 6.3 dashed green arrow), is to evaluate if
they can account for task-related variations of single movement executions [81]. In practice, one might assess the capability of these synergies to decode each repetition of different motor tasks. In other words, one should be able to classify the motor tasks from the activation coefficients of the extracted synergies. If the decoding capability is satisfactory, one might conclude that the synergies not only constitute a low dimensional, but also a functional representation of the motor commands. This idea might be used to develop novel extraction algorithms that include task decoding objectives directly in the optimization procedure. The identified synergies would then maximize not only the reconstruction of the original motor patterns, but also the capability of disambiguating task-relevant trial-to-trial variations. Unlike the dimensionality reduction methods used so far, this approach would rely on supervised learning techniques to exploit information about the task. Possible alternatives to standard extraction algorithms include energy constrained discriminant analysis [199], the discriminant NMF [200], and the hybrid discriminant analysis [201].

The use of single-trial analysis, like the decoding strategy proposed above, may be useful for addressing some open problems that are relevant to this review. First, the development of such techniques may be useful to identify muscle activation components of relatively low amplitude that reflect unique information about the task [202]; such components would be completely lost if an average across several trials is performed prior to the analysis. Second, such single-trial analysis techniques may be used to investigate the existence of trial-to-trial correlations across synergy activations, and to evaluate their functional role in controlling and performing task-related movement [203, 204]. Finally, approaches based on single-trial analysis of neural activity could also be instrumental in clarifying the existence of a neural basis for the muscle synergies [127, 133, 142, 205]. For example, they could in principle be applied to decode the task from single-trial neural population patterns that regulate the activation of synergies, and also to determine which patterns encode task differences, and which carry additional or independent information to that carried by other patterns [128].

Finally, an important aspect that is worth discussing is the role of feedback loops. In the case of synchronous synergies, the time course of the mixing coefficients can be adjusted on-line by means of appropriate feedback controllers; this is the reason of the popularity of such a model in the context of robotics. On the contrary, the models of temporal and time-varying synergies, in which the actuation time course are directly embedded in the synergies themselves, naturally represent feedforward controllers. As a result,
the evolution of the task-variables intimately depends on the initial condition of the dynamical system. Alternatively, these synergies might be defined as functions of both time and state-variables; such an approach would characterize temporal and time-varying synergies as generators of complete control policies \[167, 171, 180\].

In conclusion, we believe that the evidence reviewed here provides support for the existence of muscle synergies. However many issues are still unresolved. A deeper investigation of the relationship between synergies and task variables might help to address some of the open questions. In general, a closer coordination between experimental and computational research might lead to a more objective assessment of the muscle synergy hypothesis in task-space, and a better understanding of the modularity of the CNS.
Figure 6.2: Different models of muscle synergies. The temporal and the synchronous models explain motor signals as linear combinations of muscle balance vectors (spatial patterns), with 1-dimensional time-varying coefficients (A). In the temporal model, these coefficients serve as task-independent predefined modules, and the spatial patterns as the new (task-dependent) control input. In the synchronous model, on the other hand, the control input is represented by the temporal patterns, while the spatial patterns act as predefined modules. Finally, the time-varying synergies are spatio-temporal predefined motor patterns, which can be scaled and shifted in time by the new input coefficients.
Figure 6.3: Different procedures for the identification and the testing of muscle synergies. In experimental neuroscience (green arrows), initially a group of subjects perform the tasks prescribed by the experimenter (A). The EMG signals acquired during the experiments (B) are then analyzed, and a dimensionality reduction algorithm is applied to obtain the synergies (C). Very often such synergies are not evaluated at the task-level (dashed arrow), therefore there is no guarantee that they lead to the observed task performance. In robotics (red arrows), synergies are synthesized (C) based on the requirements of the desired class of tasks (A). Then they are appropriately combined to generate the motor signals (B) to solve a specific task instance. The quality of the synthesized synergies is finally tested in terms of the obtained task performance (A). Without loss of generality, the figure presents the time-varying synergy model; however, the previous description holds for all the models.
Chapter 7

Synthesis and Adaptation of Effective Motor Synergies via Dynamic Response Decomposition

7.1 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 muscle activity may emerge from the combination of predefined muscle patterns, the so-called muscle synergies. This organization seems to explain muscle contractions across a wide range of complex tasks (e.g. running, walking, keeping balance, reaching and other combined movements) both in humans and in animals.

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. 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 and via-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. Nori and Frezza [167] 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 [168]. Alessandro et al. [179] 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. Todorov and Ghahramani [178] employ an unsupervised learning procedure to identify muscle synergies from a collection of sensory-motor data obtained by actuating a robot with random signals. Berniker et al. [60] defines synergies as the minimal set of input vectors that influences the output of a reduced-order model of the agent, and that minimally restrict the commands (and the resulting responses) useful to solve the desired tasks [60]. Practically, this set is found by optimizing the synergies against a representative dataset of desired sensory-motor signals. Schaal et al. [206] propose the architecture of the dynamic movement primitives (DMP): 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 proposed by Todorov and Ghahramani [178]), and it provides a computationally fast method to obtain the synergy combiners 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.

7.2 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, the dynamic response decomposition (DRD): a general procedure to compute actuations that solve generic reaching tasks (see Sec. 7.2.1). Subsequently, in Sec. 7.2.2, 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

\[ D(q(t)) = u(t), \]

where \( q(t) \) represents the time-evolution of its configuration variables (their derivatives with respect to time are \( \dot{q}(t) \)), and \( u(t) \) is the actuation applied. Inspired by the hypothesis of muscle synergies\(^1\), we formulate the actuation as a linear combination of predefined motor co-activation patterns:

\[ u(t) = \sum_{i=1}^{N_\phi} \phi_i(t)b_i := \Phi(t)b, \]

(7.1)

where the functions \( \phi_i(t) \in \Phi \) are called motor synergies. The notation \( \Phi(t) \) describes a formal matrix where each column is a different synergy. If we consider a time discretization, \( \Phi(t) \) becomes a \( N \times N_\phi \) matrix, where \( N \) is the number of time steps, \( \dim(q) \) the dimension of the configuration space and \( N_\phi \) the number of synergies.

We define dynamic responses (DR) of the set of synergies as the responses \( \theta_i(t) \in \Theta \) of the system to each synergy (i.e. forward dynamics):

\[ D(\theta_i(t)) = \phi_i(t) \quad i = 1...N_\phi. \]

(7.2)

with initial conditions chosen arbitrarily.

7.2.1 Solution to reaching tasks: the dynamic responses decomposition

A generic reaching task consists in reaching a final state \((q_T, \dot{q}_T)\) from an initial state \((q_0, \dot{q}_0)\) in a given amount of time \( T \) satisfying intermediate

\(^1\)With respect to the model of time-varying synergies, in this paper we neglect the synergy onset times.
constraints called via-points. In the case of a single via-point defined at time $t_v$, the reaching task can be formalized as follows:

$$
q(0) = q_0, \quad \dot{q}(0) = \dot{q}_0,
$$

$$
q(t_v) = q_v, \quad \dot{q}(t_v) = \dot{q}_v,
$$

$$
q(T) = q_T, \quad \dot{q}(T) = \dot{q}_T.
$$

Controlling a system to perform such tasks amounts to finding the actuation $u(t)$ that fulfills the point constraints $^2$. Specifically, assuming that the synergies are known, the goal is to identify the appropriate synergy combiners $b$. In this paper we consider both point-to-point reaching tasks (i.e. with no via-point constraints) and via-point reaching tasks.

The procedure consists of, first, solving the problem in kinematic space (i.e. finding an appropriate $q(t)$), and then computing the corresponding actuations. From the kinematic point of view, solving the task can be seen as an interpolation problem; i.e. a set of functions is used to generate a trajectory $q(t)$ that interpolates the constraints defined in (7.3). To build this interpolant one could use orthonormal polynomials, trigonometric functions or Gaussian, to mention just a few of the many options. Herein, this set is composed by the dynamic responses of the synergies (the quality of the DRs as building blocks for interpolants is evaluated in Sec. 7.3):

$$
q(t) = \sum_{i=1}^{N_\theta} \theta_i(t) a_i := \Theta(t) a. \quad (7.4)
$$

As mentioned earlier, if time is discretized, $\Theta(t)$ becomes a $N \dim(q)$-by-$N_\theta$ matrix, where $N_\theta$ is the number of dynamic responses. The vector of combiners $a$ is chosen such that the task constraints are satisfied (i.e. the task is solved). Specifically, this vector is computed by solving the following linear system of equations:

$$
\begin{pmatrix}
\theta_1(0) & \ldots & \theta_{N_\theta}(0) \\
\theta_1(t_v) & \ldots & \theta_{N_\theta}(t_v) \\
\theta_1(T) & \ldots & \theta_{N_\theta}(T) \\
\dot{\theta}_1(0) & \ldots & \dot{\theta}_{N_\theta}(0) \\
\dot{\theta}_1(t_v) & \ldots & \dot{\theta}_{N_\theta}(t_v) \\
\dot{\theta}_1(T) & \ldots & \dot{\theta}_{N_\theta}(T)
\end{pmatrix}
\begin{pmatrix}
a_1 \\
a_2 \\
a_3 \end{pmatrix} =
\begin{pmatrix}
q_0 \\
q_v \\
q_T \\
\dot{q}_0 \\
\dot{q}_v \\
\dot{q}_T
\end{pmatrix}.
$$

The matrix $M$ in the left-hand side is called *alternant matrix*, and the solvability of the problem depends on its rank. If the matrix has full row rank,
any point constraint can be solved. Otherwise, the possibility to find an exact solution (as opposed to an approximation) becomes strictly dependent on the specific task. In general, if the rank of the alternant matrix (not necessarily equal to number of rows) is equal to the rank of the augmented matrix $[M|P]$, where $P$ is the point constraint vector, the specific problem can be solved exactly. Section 7.3 presents some examples of such a situation. This observation, and its implications for the hypothesis of muscle synergies, are further discussed in Sec. 7.4.

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. $\mathcal{D}(\Theta(t)a) = \dot{u}(t)$. Finally, the vector $b$ can be computed by projecting $\dot{u}(t)$ onto the synergy set $\Phi$. If $\dot{u}(t)$ does not belong to the linear span of $\Phi$, the solution can only be approximated in terms of a defined norm (e.g. Euclidean):

$$b = \text{arg min}_b \|\dot{u}(t) - \Phi(t)b\|. \quad (7.6)$$

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

$$\Phi^+\dot{u} = \Phi^+\mathcal{D}(\Theta a) = b. \quad (7.7)$$

This equation highlights the operator $\Phi^+ \circ \mathcal{D} \circ \Theta$ (denotes operator composition) as the mapping between the kinematic combiners $a$ (kinematic solution) and the synergy combiners $b$ (dynamic solution). Generically, this operator represents a nonlinear mapping $F: \mathbb{R}^{N_\theta} \rightarrow \mathbb{R}^{N_\phi}$, and it will be discussed in Sec. 7.4.

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

Interpolation error: measures the quality of the interpolant $\Theta(t)a$ with respect to the task. Strictly speaking, only the case of negligible errors corresponds to interpolation; a non-zero error indicates that the trajectory $\Theta(t)a$ only approximates the task. The interpolation error is formulated as

$$\text{err}_I = \sqrt{\sum_{k \in K} e_{I_k}^2}$$

$$K = \{0, v_1, \ldots, v_n, T\}$$

$$e_{I_k}^2 = |q_k - \Theta(t_k)a|^2 + |\dot{q}_k - \dot{\Theta}(t_k)a|^2$$

(7.8)

where $\|\cdot\|$ denotes the Euclidean norm, and the difference between angles are mapped to the interval $(-\pi, \pi]$. The subindex indicates the point constraint, i.e. $k = 0$ for the initial condition, $k = v_i$ for the i-th via-point, and $k = T$ for the final condition. In this work we consider tasks with a single or
Projection error: measures the distance between the actuation $\tilde{u}(t)$ that solves the task, and the one obtained by the linear combination of the synergies $\Phi$

$$\text{err}_P = \sqrt{\int_0^T ||\tilde{u}(t) - \Phi(t)b||^2 dt}. \quad (7.9)$$

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

$$\text{err}_F = \sqrt{\sum_{k \in K} e_{Fk}^2}$$

$$K = \{0, v_1, \ldots, v_n, T\}$$

$$e_{Fk}^2 = ||q_k - \tilde{q}(t_k, \lambda)||^2 + ||\dot{q}_k - \dot{\tilde{q}}(t_k, \lambda)||^2 \quad (7.10)$$

Replacing $\tilde{q}(t_k, \lambda)$, $q_k$ and $\dot{q}_k$ with their corresponding end-effector values provides the forward dynamics error of the end-effector.

### 7.2.2 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 $\Phi_0$ in order to obtain the corresponding DRs $\Theta_0$. The reduction phase consists in solving a small number of reaching tasks in kinematic space (that we call proto-tasks) as described in Eq. (7.4) and (7.5); the elements of the set $\Theta_0$ are used to perform the interpolation. These solutions are then taken as the elements of the reduced set $\Theta$. Finally, the synergy set $\Phi$ is computed using relation (7.2), 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 such synergies may capture essential features of both the proto-tasks and the dynamics of the system, and therefore they might be useful to solve similar tasks. If the system starts from the same initial conditions, the actuations needed to solve such tasks will resemble the ones required by the proto-task, therefore they might be approximated by a linear combination of the synergies. Thus, we expect that the more similar the proto-tasks are to the tasks to be solved (in terms of Eq. (7.3)), the lower the errors they will induce. Section 7.3 provides some examples.

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 reaching tasks, the proto-tasks should cover relevant regions
of the state space. 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.

7.3 Results

We apply the methodology described in Sec. 7.2 to a simulated planar kinematic chain (see [207] for model details) modeling a human arm [101]. In the exploration phase, we employ an extensive set of motor signals $\Phi_0$ to actuate the arm model and generate the corresponding dynamic responses $\Theta_0$. The panels in the first row of Fig. 7.1 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).

![Figure 7.1: 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.](image)

In order to evaluate the validity of the general method described in
Sec. 7.2.1, we use the sets $\Phi_0$ and $\Theta_0$ to solve 13 different reaching tasks without performing the reduction phase (see Supplementary Material for the target locations). The second row of Fig. 7.1 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 7.1. The results are highly satisfactory for both 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 $a$ and $b$ equals the number of actuations used in the exploration.

<table>
<thead>
<tr>
<th></th>
<th>Min. Jerk</th>
<th>Random</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\text{err}_I$</td>
<td>$10^{-15}$</td>
<td>$10^{-15}$</td>
</tr>
<tr>
<td>$\text{err}_P$</td>
<td>$10^{-5}$</td>
<td>$10^{-3}$</td>
</tr>
<tr>
<td>$\text{err}_F$</td>
<td>$10^{-4}$</td>
<td>$10^{-3}$</td>
</tr>
</tbody>
</table>

Table 7.1: Order of the maximum errors obtained by using $\Phi_0$ and $\Theta_0$ (no reduction phase).

The objective of the reduction phase is to generate a small set of synergies and DRs that can solve desired tasks effectively. As described in Sec. 7.2.2, this is done by solving a handful of proto-tasks that are representative examples of the motor tasks to be solved. In the following subsections we show the details of the reduction phase for the cases of point-to-point and via-point reaching tasks as well as the obtained results.

### 7.3.1 Point-to-point reaching tasks

Point-to-point reaching tasks are formalized by a set of four constraints that prescribe desired initial and final joint angles and velocities. In this section we restrict our analysis to the subclass of tasks that impose motionless initial and final postures, i.e. $\dot{q}_T = \dot{q}_0 = 0$, and that are characterized by the same initial posture $q_0$ (see Supplementary Material). Since the proto-tasks that we employ adhere to the same restrictions, the linear system in Eq. (7.5) becomes:

$$
\begin{pmatrix}
q_0 & \cdots & q_0 \\
\theta_1(T) & \cdots & \theta_{N_y}(T) \\
0 & \cdots & 0 \\
0 & \cdots & 0 \\
\end{pmatrix}
\begin{pmatrix}
a \\
q_0 \\
q_T \\
0 \\
\end{pmatrix} =
\begin{pmatrix}
e \\
0 \\
0 \\
\end{pmatrix},
$$

(7.11)

where $\theta$ are the reduced DRs, and $q_T$ is the final desired posture as prescribed by a given task. Note that each element of the extended matrix is a two-
dimensional column vector. Therefore, such a matrix consists of 4 non-zero rows, of which the first two have the same elements in each column. This formulation shows that, from the kinematic point of view, any problem of this class can be solved if the rank of the alternant matrix is equal to 3, which implies that at least three DRs are needed. However, the number of synergies required to approximate the corresponding actuations might be higher.

Figure 7.2 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 $\Phi_0$, see Tab. 7.1) 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. The exact locations of the proto-tasks are provided in the Supplementary Material section.

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 7.3 shows this comparison. The tasks consist in reaching the 13 targets in Fig. 7.1 (see Supplementary Material for the exact target locations). 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, comparable with the 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 and effective set of synergies out of an extensive set of actuations. It is noteworthy that the 6 reduced DRs lead to an alternant matrix with rank equal to 3, therefore any point constraints of this kind can be solved exactly and the obtained interpolation error is negligible for all the testing tasks. Due to the nonlinearity of the dynamical system, the actuations corresponding to the computed kinematic solutions can only
Figure 7.2: 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).
be approximated, leading to higher projection errors. However, the reduced synergies can generate approximations that lead to high task performances, i.e. forward dynamics errors in the range \([10^{-3}, 10^{-2}]\), satisfactory for many ordinary tasks and greatly reducing the dimensionality of the controller.

Figure 7.3: Evaluation of the reduction phase for the point-to-point tasks. Errors produced by subsets randomly selected from the exploration-actuations (boxplots) are compared with the errors obtained after the reduction phase (filled circles).

### 7.3.2 Via-point reaching tasks

The via-point tasks considered here, including the proto-tasks, are characterized by zero velocity at each point constraint, i.e. \(\dot{q}(0) = \dot{q}(t_v) = \dot{q}(T) = 0\), and by the same initial and final position placed in the center of the operational space, i.e. \(q(0) = q(T) = q_0\) (see Supplementary Material). In other words, the agent is required to reach a certain location with zero velocity (i.e. the via-point), and return to its initial posture (such movements are sometimes described as reversal movements). These tasks have relevance as they resemble the motion typically performed for carrying objects to and from the agent, e.g. reaching for food and bringing it to the mouth or picking up a salient object and bringing it for closer examination. Under these conditions, the linear system of equations (7.5) used to compute the kinematic solution, now becomes:

\[
\begin{pmatrix}
    q_0 & \cdots & q_0 \\
    \theta_1(t_v) & \cdots & \theta_{N_p}(t_v) \\
    q_0 & \cdots & q_0 \\
    0 & \cdots & 0 \\
    0 & \cdots & 0 \\
    0 & \cdots & 0
\end{pmatrix}
\begin{pmatrix}
    a
\end{pmatrix} = 
\begin{pmatrix}
    q_0 \\
    q_v \\
    q_0 \\
    0 \\
    0 \\
    0
\end{pmatrix}.
\tag{7.12}
\]

For the same rationale discussed in Sec. 7.3.1, to guarantee the existence of an exact kinematic solution for any via-point task of this kind, the rank of
the alternant matrix, and therefore the minimal number of DRs, should be equal to 3. However, the number of synergies required to obtain satisfactory values of projection and forward dynamics errors might be higher.

The proto-tasks are chosen by means of the same procedure employed in the case of point-to-point reaching tasks. In this case, since the position of the hypothetical object to be picked up is unknown, the location of the via-point is placed in the region of the operational space with the highest projection error. As shown in Fig. 7.4, this strategy aims at decreasing the projection error over the entire end-effector space, such that eventually the actuations necessary to solve any via-point task of this class can be approximated satisfactorily. The locations of the 9 via-points are provided in the Supplementary Material section.

Figure 7.5 shows the evolution of the average projection error as a function of the number of proto-tasks. Depending on the precision required, more or less proto-tasks can be used. Here we employ 9 proto-tasks to obtain an average projection error of around $10^{-2}$. This implies that the actuations to solve this class of tasks can be approximated by combining only 9 synergies. It is important to stress that, due to the nonlinearities of the dynamical system, the projection error serves only as an heuristic estimate of the actual error made when executing the task; the latter is directly quantified by the forward dynamics error, the distribution of which is depicted in the bottom right panel of Fig. 7.4.

Like in the case of point-to-point reaching tasks, the reduced synergies are compared to 100 randomly chosen subsets of 9 actuations drawn from the exploration set of low-passed random motor signals. The tasks consist in reaching the 13 targets shown in Fig. 7.1 with zero velocity and going back to the original posture (initial and final velocities should also be zero). The results shown in Fig. 7.6 provide additional evidence that the reduction phase identify effective synergies: the mean error of the random subsets (boxplot) is about 2 order of magnitude higher than the error corresponding to the reduced synergies (filled circles), and the forward dynamics errors lie in the range $[10^{-3}, 10^{-2}]$, meaning that the 13 approximated actuations lead to good task performance. Such observations confirm the conclusion developed in the case of point-to-point reaching targets that the reduction is able to build a small set of effective synergies out of the large exploration set.

### 7.3.3 Reduction vs. PCA

A legitimate question is whether the principal components (PC) of the exploration set of actuations render a useful set of synergies. The results in Fig. 7.3 and 7.6 already indicate that a linear transformation of the explo-
Figure 7.4: Selection of via-points 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 the via-point in that position and to go back to the original posture. The bottom right diagram shows the forward dynamics error of the end-effector using 9 proto-tasks (9 synergies).
Figure 7.5: Average projection error as a function of the number of proto-tasks for the via-point reaching tasks.

Figure 7.6: Evaluation of the reduction phase for the via-point tasks. Errors produced by subsets randomly selected from the exploration-actuations (boxplots) are compared with the errors obtained after the reduction phase (filled circles).
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. 7.3 and 7.6). If the agent finds an acceptable solution to a proto-task, such a solution is used to generate a new synergy (populating the set $\Phi$), 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; Fig. 7.2 and 7.4 illustrates this mechanism. As it can be seen, the development of the synergy set incrementally improves the ability of the agent to perform 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 ability 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. Similar results have been reported using other nonlinear differential operators besides kinematic chains [208]. These are not trivial results since the reduced synergies over-perform the subsets randomly taken from the exploration set \( \Phi_0 \) (see Fig. 7.3). 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 several nonlinear dimensionality reduction methods. This verification goes beyond the scope of this paper.

An important aspect of our method is the relation between \( \Theta \) and \( \Phi \) (see Eq. (7.2)). This mapping makes explicit use of the body parameters (embedded in the differential operator \( D \)), 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 [79, 154, 155], biomechanical constraints are not explicitly included in the extraction algorithms. Additionally, Eq. (7.2) 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 Nori [168] and Alessandro et al. [179], these observations highlight the importance of the body in the hypothetical modularization of the CNS.

A direct consequence of the nonlinearity of the body dynamics is that small variations of the actuation might lead to quite different kinematic trajectories. This justifies the need to distinguish between projection and forward dynamics error, and it clarifies why the former can be considered only as an heuristic estimate of the error of the synergy based controller, which is more directly evaluated in terms of the forward dynamics error. While many studies in experimental neuroscience solely analyze the validity of the muscle synergy hypothesis based on the capability of a set of generators to reconstruct various dataset of recorded muscle activations [68, 94, 96, 107, 108], we believe that the introduction of complementary evaluations that include task-based measures could better elucidate the modularity of the CNS.

In this vein, some researchers introduced the concept of functional synergies: the components of a dataset that includes both muscle activations and measurements of specific task variables (e.g. joint angles, end-limb force) [80, 90]. As a result, each component consists of a pattern of muscle contractions and an evolution of such variables (that represents the corresponding biomechanical functionality). The observation of the same components over different tasks is taken as an indirect evidence for the existence of neuromechanical motor modules. Such an approach is not too different from the idea behind DRD: each synergy corresponds to its DR (i.e. biomechanical
functionality), and the solution of a given task is obtained by their linear combinations. However, the identification of functional synergies by means of non-negative matrix factorization (NMF), implies that muscle synergies and their biomechanical functionalities are scaled by the same coefficients. This contrasts with our theoretical results that show a nonlinear relationship (the mapping $\mathcal{F}$) between the mixing coefficients of the synergies and the ones of the DRs. This observation casts some doubts on the generality of the extracted functional synergies across different motor tasks, and more generally on the validity of linear approaches (as functional synergies) to validate the hypothetical modularity of the CNS.

The mathematical formulation of the DRD, and in particular the system of linear equations (7.5), allows a series of observations about the minimum number of synergies that are required to solve a class of tasks. First, it clearly shows that the number of synergies strictly depends on the number of constraints required by the class of tasks. The higher the number of constraints, the higher the minimum number of proto-tasks (hence of synergies) necessary to guarantee the existence of an exact solution for any task in the class, i.e. full row-rank alternant matrix. Second, a highly specified class of tasks reduces the minimum number of required synergies. For example, point-to-point tasks starting from a specific initial condition and with zero initial and final velocity leads to a minimum of 3 synergies instead of 8, as would be required for a general point-to-point reaching task (see Sec. 7.3.1). It is important to stress that these considerations provide the necessary condition to guarantee the existence of kinematic solutions to any task that belong to the desired class. If the agent was only interested in some specific instances of such tasks (e.g. reaching some given points of the phase space), the minimum number of synergies can be reduced even further. Having this in mind, it is not surprising that our framework predicts a higher number of synergies than statistical analyses of biomechanical signals: those datasets are indeed recorded during specific and constrained task instances. Finally, in our method, the minimum number of synergies is computed solely based on kinematic considerations, but it does not guarantee low values of projection and forward dynamics error. In fact, as shown in Sec. 7.3, the number of synergies that is required to obtain satisfactory performances is certainly higher than the theoretical kinematic-based estimation.

In the DRD method, 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 $\mathcal{F}$ on the vector $\mathbf{a}$, hence obtaining the synergy combinators $\mathbf{b}$. Since $\mathcal{F}$ 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 $\mathcal{F}$, the input-output data generated during the exploration phase (i.e. $\Phi_0$ and $\Theta_0$) 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: an initial guess of the synergy set can be iteratively modified until convergence.

Finally, the current formulation of the method does not includes 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\cite{209} and the concatenation of solutions.

### 7.5 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 and via-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 $\mathcal{F}$ 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.

The software used to produce all the results reported in this paper is available as an Octave package under free and open source license. The reader is encouraged to download, test, report bugs and submit improvements to the algorithm.

Chapter 8

Synthesising a Reduced Dimensional Oculomotor controller

8.1 Introduction

The eye is one of the most extensively studied systems in the human physiology. Aside from an long-enduring interest in visual perception going back to the earliest days of science, the oculomotor system also serves as an interesting model in the study of biological motor control abilities; aspects such as the relative simplicity and abundance of anatomical and neurophysiological data render it a fascinating subject for motor neuroscience. Gaining an understanding of oculomotor control is not only relevant to the biologists but is also of interest to control engineers as well. However there are numerous open questions that still remain unanswered, such as what mechanism leads to the stereotypical behaviour? How is this control mechanism synthesised? And, do optimality principles underlie their synthesis and how?

It has been hypothesised that the natural extension of the requirement for high evolutionary fitness in species is to exhibit an optimisation of behaviour, during the life time of the creature, dictated by some form of behavioural cost. Examples of these cost minimisation strategies in motor control include, the Minimum-Jerk, Minimum-Torque, Minimum-Variance, Minimum-Control effort or Optimal-feedback. Although evidence has been presented justifying each case, current focus is on achieving consensus in defining what is the goal of the optimised behaviour. Nevertheless, an open problem intrinsic to all of those models, is the definition of how the optimisation might occur in learning motor control.
It is widely recognised that optimisation schemes are rendered intractable with increase in dimensionality of the system \cite{217}; an unavoidable circumstance in biological systems given the natural redundancy in the motor systems. This observation known as the eponymous Bernstein’s Problem \cite{62} gave rise to hypotheses on some form of motor coordination being necessary for behaviour. In particular if optimisation occurs over internally stored dynamic models \cite{218} One approach is to seek a reduced dimensional representation of a system; these representations can then be utilised for a control or optimisation scheme. This principle underlies the theory of motor primitives and muscle synergies \cite{64, 68, 74}. It is however not clear if such schemes could be acquired by a developmental process.

In this context, recent work on the role of spontaneous motor activity in the ontogenetic self-organisation of sensorimotor reflexes \cite{219, 220} might be relevant. In particular, this process seems to be vital for the self-organisation of oculomotor control as well \cite{221}, highlighting the importance of sleep in development of control. Although it's known that saccadic eye movements are regulated at the Brainstem \cite{222} the mechanisms underlying its development are unknown. A key consensus that seems to be emerging is that ontogenetic processes, which self organise sensorimotor relations, function on the principle of \textit{pruning} of neural connections; i.e. some form of complexity reduction seems to be an essential feature.

In this paper, we demonstrate how a reduced dimensional control of the eye may be acquired from an approach inspired by spontaneous motor activity in newly born rats; our framework utilises linear system identification and balanced reduction. We posit that this learning process could in principle occur prenatally though proprioceptive feedback and is thus indicative of developmental mechanisms underlying oculomotor control. We then utilise the model for the synthesis of a motor primitive strategy for the control of saccadic and fixation behaviours. This not only allows generation of control signals in a lower dimensional space but the resulting reduced dimensional model can be utilised to generate optimal control signals satisfying an alternative cost such as minimum-noise; an optimal bang-bang approach is utilised for generating the saccades and a steady state control strategy is used for fixation - comparing favourably to minimal-control norm approaches. Lastly, the framework is used to analyse the role of orbital tissue damping.

This paper is organised as follows. Section \ref{sec:control_framework} introduces the proposed control framework and its components including Balancing Truncation, along with a description of the linear eye model that we test our approach on. The proposed mechanism is then presented in Section 3 and results are presented in Section 4. This is finally followed by the conclusions and discussions in Section 5.
8.2 Methods

The methodology for our control synthesis is depicted in Fig. 8.1. It largely follows up from the methodology introduced in [223]. The various component steps are described in this Section.

Figure 8.1: Motor Primitive inspired Oculomotor Control Architecture. The blue arcs denote spontaneous activity and identification related process, and the red arcs denote controlled behaviours. A - The eye plant is first perturbed by different forms of Spontaneous Motor Activity. B - the resulting state trajectory is then used for linear system identification and model reduction by Balanced Truncation. C - Motor Primitives are then synthesised by the technique of [223]. Lastly, controllers for Saccade and Fixation use the reduced order model and motor primitive for control.

8.2.1 Simplified Linear Model of Oculomotor Dynamics

A number of theories of Oculomotor control use the simplification of the linear model of the eye plant. In this work we employ the following simplified linear model of horizontal eye motion is presented below; the muscles described are the lateral and medial rectii. The model presented in Fig 8.2 is described using Eq. 8.1.

131
The system equations are as follows.

\[ I \ddot{\phi} = -k_t \phi - b_t \phi + \vec{f}_{ml} + \vec{f}_{mm}, \]
\[ \vec{f}_{ml} = \vec{f}_{pi} + \vec{f}_{ai}, \quad \vec{f}_{pi} = -k_m \phi, \quad \vec{f}_{ai} = C_m \alpha_i, \]
\[ \dot{\alpha}_l = \frac{1}{\tau}(u_l - \alpha_l), \quad \dot{\alpha}_m = \frac{1}{\tau}(u_m - \alpha_m), \]  

(8.1)

where \( m \) is the mass of the eyeball at position \( \phi \), and it is acted upon by forces from the lateral rectii \( f_{ml} \), and the medial rectii \( f_{mm} \) apart from passive forces of the orbital tissue represented by the linear stiffness \( k_t \), and damping \( b_t \). The force generated by each of these rectii \( i \) consists of a passive component \( f_{pi} \) due to the passive stiffness in the muscle \( k_m \) and an active part \( f_{ai} \) which is proportional to the activation \( \alpha_i \) by the constant \( C_m \). The muscle activation considered is of first order dynamics depending on the time constant \( \tau \) and the inputs to the whole system are through \( u_l \) and \( u_m \). The system is thus fourth order with input dimensionality being 2. The output in this case is the eye horizontal position \( \phi \).

This kind of system is rewritten into a state-space linear system form as,

\[ \dot{x} = Ax + Bu, \quad y = Cx, \]

(8.2)

where, \( u \in \mathbb{R}^I \) is the input, \( x \in \mathbb{R}^N \) is the state, \( y \in \mathbb{R}^O \) is the output. The matrices \( A, B, \) and \( C \) are commonly called as the state, input, and output matrices, respectively.
8.2.2 Spontaneous Motor Activity

In mammals, the process of spontaneous motor activity (SMA) carries out muscle contractions in the absence of sensory stimulation, i.e. during sleep. This type of motor activity has been observed during sleep throughout all developmental stages (including the foetal stage) [224]. One particular type of SMA, the Myoclonic twitch, spontaneously triggers independent contractions of individual muscles. This activity seems to have a major role to play in the ontogenetic self-organisation of sensorimotor reflexes [219, 220]. This process has also been observed in the oculomotor system with EMG measurements and is hypothesised to drive REM sleep.

Inspired by this process, we utilise short (< 1.0 sec) pulses with and without co-contraction or reciprocal innervation, as seen in Fig. 8.3b to perturb the mechanical system, resulting in a motion output that can be recorded in the form of a dataset (see block (A) in Fig 8.1) of snapshots of the dynamical system as $\chi = [x(t_0), \ldots, x(t_i)]$, where $\chi \in \mathbb{R}^{4 \times n_t}$ and $x(t_i)$ is the $n_t$th snapshot of the system, where $n_t$ is the total number of snapshots in the dataset (or datapoints) and the state dimensionality is 4.

8.2.3 Dimensionality Reduction by Balancing

The projection framework for dimensionality reduction is so named because the reduced dimensional model that is sought is obtained by projecting the dynamics of the source system into a lower dimensional sub-space. Balanced Reduction refers to a family of projection based dimensionality reduction techniques that are based on first obtaining a balanced system representation; a representation where the state variables are organised on the basis of their importance for controllability and observability (i.e. on their importance for control). Although the technique is briefly introduced here, further information may be found in [225] or in the references within [67]. While the original algorithm is a technique for reduction of linear systems, variations for nonlinear systems exist as well [226].

The technique first obtains a balancing transformation $T$, defined by,

$$\tilde{x} = Tx,$$

which rotates the state variables on the basis of controllability grammian ($P$) and observability grammian ($Q$) [225], thus ranking them on their relative importance. The actual rank is on the basis of their Hankel Singular Values (HSV), defined by the product of the two grammians as,

$$\Sigma = \sqrt{P \times Q},$$
where the resulting set $\Sigma$ are called the HSV, and are ordered by $\sigma_1 \geq \sigma_2 \ldots \sigma_k$.

Once the rotated representation is obtained, two kinds of approaches can be employed. One technique, is to simply truncate the system by only considering the most important state variables (ranked by the HSV values) - balancing truncation, and the second is a variation where the least important state variables are set to their steady state values - singular perturbation. In this paper, we utilise both these techniques to reduce the dimensionality. In either case, the reduced order model is obtained as,
\[ \dot{x}_k = A_k x_k + B_k u, \quad y = C_k x_k + D_k u, \quad (8.5) \]

where the obtained dimensionality is \( A_K \in \mathbb{R}^1, B_K \in \mathbb{R}^{2 \times 1}, C_K \in \mathbb{R}^1, D_K \in \mathbb{R}^{2 \times 1} \).

It must be noted that balanced reduction methods are not inherently data driven, and utilise a closed form solution computed from the source model. In this paper, we therefore utilise the approach of first utilising linear system identification using least-squares and then follow it up with balancing as presented in the methodology in Fig. 8.1.

### 8.2.4 Biological Motor Primitives

The concept of spinally stored motor primitives \cite{64, 227} were proposed on the basis of experiments on microstimulation of the spine of the frog \textit{Rana Pipiens}. Stimulation of a few spinal regions resulted in the movements of leg towards unique equilibrium positions; this model of control is hence called as Spinal Force Fields. The related principle of Muscle Synergies \cite{68} presents a model of stored muscle patterns which are suitably linearly combined by spinal circuits to generate motor behaviours for various tasks.

A recent work on a model of motor primitives\cite{60, 161} introduced a formalisation of spinally stored constraints on the motor input commands of the form,

\[ u = U^* W, \quad (8.6) \]

where, \( U^* = u^*_{1..k} \) is a set denoted as the motor primitives, comprising of \( k \) primitives, and \( W \) is the vector of reduced dimensional control inputs, \( C = [w_1, \ldots, w_k]^T \), which can be thought of as “weights” combining the primitives linearly. Each column of \( U^* \) can be thought of as a set of spinally stored muscle activations, similar to the activations produced by microstimulation of the frog’s spine. In this formulation, each primitive \( u^*_i \) is in the dimension of total number of muscles present\cite{60}, while only \( k \) primitives are needed and used to specify their motion. In this form, the primitives represent the basis vectors of the desired space of motor commands.

The computation of the motor primitives \( U^* \) is through optimising a cost against a synthesised reduced order model of the system; the cost is based on the new space of control commands being \textit{non-negative} (real muscles cannot be negatively activated), \textit{orthogonal} (act independently) and \textit{useful} for generating commands (formalised based on mathematical properties of the equivalent reduced dimensional system).

A modified framework was proposed based on learning primitives using a Singular Value Decomposition (SVD) instead of numerical optimisation
This latter approach is utilised in this paper. This is defined as,

\[ U^* = V_{1..k}, \] (8.7)

where \( V_{1..k} \) are the first \( k \) columns of \( V \), the right singular matrix obtained from the svd of the reduced order input matrix \( B_k \) as, which is decomposed as, \( B_k = U \Sigma V \). Utilising the components of the right singular matrix ensures that the primitives that obtained are useful, since the vectors are orthogonal and the new inputs \( W \) span the reduced dimensional space \( W \in \mathbb{R}^k \), where in this case, \( k = 1 \).

### 8.2.5 Motor-primitive based Oculomotor control

We now present a 1D saccade and fixation controller which computes the optimal input \( W \).

#### Time-Optimal Primitive based Saccadic Control

One kind of model of saccadic control to cite bang-bang-control is the minimum-time control condition which is achieved optimally using a bang-bang input \( W(t) \) defined by,

\[
W(t) = \begin{cases} 
\pm W_{\text{max}}, & \text{if } t \leq t_{\text{sac}} \\
W_{\text{fix}}, & \text{if } t > t_{\text{sac}}
\end{cases}
\] (8.8)

The optimal solution is to control the timing of the pulse \( t_s \) for achieving a saccade to a desired position \( x_d \). To derive this, first we solve the higher order non-homogenous matrix differential equation of Eq.8.2 for a step input \( (W_{\text{max}}) \). This is given by,

\[
x(t) = e^{At}x_0 + \int_0^t e^{A(t-\tau)}BU^*Wd\tau,
\] (8.9)

integrated until the end of the saccade period \( t_s \) as,

\[
x(t_s) = -[I - e^{At_s}]A^{-1}BU^*W_{\text{max}}.
\] (8.10)

The optimal time \( t_s \) minimises the norm,

\[
\arg \min_{t_s} f(t_s) = \| x(t_s) - x_d \| \] (8.11)

Note that the solution computed in Eq.8.10 can utilise the minimum dimensional components \( A_k \) and \( B_k \), thus utilising the learned minimum dimensional model to compute an optimal control. The minim dimensional components are also utilised for a Fixation controller explained subsequently.
8.2.6 Minimum Dimensional Primitive based Fixation Control

The fixation controller ensures that the eye maintains the position at equilibrium. Using the primitives computed in For the obtained system in Eq. 8.5, the equilibrium state for a given input corresponds to $\dot{x}_k = 0$, which is therefore,

$$x_k = -A_k^{-1}B_k u, \quad y = \left[ -A_k^{-1}B_k + D_k \right] u. \quad (8.12)$$

Since the input $u$ is constrained according the motor primitive as in Eq. 8.6, it is sufficient to compute the required reduced dimensional control inputs $W$ for a desired state $x_d$. This is derived as,

$$W_d = \left[ \left(-A_k^{-1}B_k + D_k \right) u^* \right]^{-1} x_d. \quad (8.13)$$

Note that if $k$ is chosen to be of same dimensionality of the output $y$, Eq. 8.13 is computed using a regular inverse instead of the pseudo-inverse and thus the redundancy problem is directly resolved.

8.3 Experiments and Results

The first phase consists of an unsupervised training using a perturbation signal to learn a suitable lower dimensional representation of the behaviour of the system. This is then used to compute controllers to carry out saccadic tasks.

8.3.1 Control of Saccading and fixation

8.3.2 Effect of Orbital Tissue Damping

The effect of Orbital Tissue Damping on the reduced dimensionality was analysed by repeating the methodology for 10 values of damping uniformly distributed in the range $[0.0001, 0.9]$ deg/sec. Controller performance in each case was analysed for the 10 targets in the range $[10^\circ, 30^\circ]$. The performance was quantified on the basis of mean error in reaching the targets. The figure shows the response for a step command over a much longer interval of 15 seconds.

8.4 Discussion

This form of control works on projection of dynamics. The notion of minimisation of “effort” in the form of a norm does not have a biological basis.
Figure 8.4: Saccades with an optimal bang bang reduced dimensional controller using motor primitives - results for a range of orbital tissue damping values. (a) The trajectories of the eye for a set of saccadic targets in the range $[10^\circ, 30^\circ]$ indicated by blue lines, the coloured lines indicate trajectories of the eye for various orbital tissue dampings, (b) The optimal saccade durations computed for each case.

In the controller we demonstrate, internal models for control can be in the reduced dynamics space. This does have some biological plausibility and shall be expanded further in future works. While the reduced model can be learnt in an unsupervised learning process prenatally, the controller can be learned on the basis of a supervised learning process. The model we propose can be used to predict morphological properties of the biological oculomotor system.
Figure 8.5: Plot of peak saccade velocity against saccadic magnitude - a main sequence imposed by the optimal controller, for various values of orbital tissue damping.

Figure 8.6: Step responses for a range of reduced dimensional models acquired against various orbital tissue dampings. The greater the damping, the greater the time taken to settle.
Figure 8.7: Comparison of steady state errors without the optimal saccade controller and with, (a) Steady state error after 15 seconds using a reduced dimensional steady state control step, (b) Steady state error after 1 second using a reduced dimensional optimal bang-bang controller - errors are slightly decreased
Chapter 9

Spine dynamics as a computational source in spine-driven quadruped locomotion

9.1 Introduction

In conventional robot design, rigid body and high torque servos are widely used to get precise control and to suppress unwanted dynamics. Although this control architecture allowed a robot to achieve multiple tasks, it has required much computational cost because it has to control every degree of freedom precisely all the time. Recent studies inspired from biological systems have shown that a compliant physical body, featuring nonlinear properties, has the potential capabilities to achieve some tasks, e.g., locomotion \[228\]. This indicates that partial computation of control can be outsourced to the body by using material properties or other morphological properties.

This hypothesis, refer to as morphological computation, has recently received some theoretical support by Hauser et al \[192\]. This theoretical model takes a compliant physical body as a reservoir and states that this compliant physical body can be a potential computational resource, due to the elasticity and nonlinearity embedded in this physical body. Furthermore, to achieve autonomous generation of adaptive periodic patterns, e.g., locomotion, this theoretical model states that the introduction of a static feedback into morphological computing system with compliant bodies is necessary \[229\].

In this study, we applied this theoretical model to a robot named Kitty. Kitty robot is a spine-driven quadruped robot and features a biologically
inspired, compliant, actuated multi-joint spine. We found that the dynamics of this spine has a specific correspondence to each behavior such as bounding, trotting, and turning behavior. This suggests the possibility of this compliant spine to be a diverse behavior generator. Therefore, our emphasis is laid on spinal structure as a computational source to achieve motion primitives, for example, in achieving gait generation and diverse locomotion tasks, by introducing a linear, static feedback to the body reservoir.

In the following sections, we first explain the possibility of Kitty robot developed to investigate how the dynamics of the spine can be regarded as a computational source. Then, the information processing is described, followed by experimental results by using actual sensory data from real robotic system. The results suggest that this computational source (compliant spine) is able to encode movement patterns, produce rhythmic patterns, learn new pattern.

9.2 Experimental Setting

9.2.1 Biologically inspired compliant spine

We use a spine-driven quadruped robot called Kitty (Fig. 9.1(a)) as a testbed to investigate how the dynamics of the spine can be regarded as a computational source which is able to generate various patterns. Kitty is equipped with a flexible actuated spine (29 cm wide, 32 cm long, 20 cm high, and 1.4 kg) (Fig. 9.1(a)). Fig. 9.1 (b) shows an artificial spine endowed with biological characteristics. It consists of cross-shaped rigid vertebrae made of ABS plastic, intervertebral disks made of silicone blocks and strings driven by motors, similar to the anatomical spine structure. The vertebrae are separated by the silicon blocks, which work as intervertebral discs, and connected by four strings through themselves and the silicon blocks. The four strings connecting vertebrae and intervertebral disks are pulled respectively by four RC motors, which can control the movement of the spine (Fig. 9.1 (c)).

Motor command \( I_i(t) \) to motor \( i \) for the spine movement is computed with sinusoid function given by:

\[
I_i(t) = A \sin(2\pi f_i t + \phi_i) + \psi \quad i = \{d, l, r, u\},
\]

where \( i \) stands for the position of motor embedded in the robot. \( d, l, r, \) and \( u \) indicate motors controlling the strings located at lower, left, right and upper position, respectively. The dynamics of the spine is captured by force sensors (FSR400) randomly embedded into silicone blocks, as shown in Fig. 9.1 (d).
Figure 9.1: (a) A quadruped robot equipped with a tendon-driven spine. (b) A biologically inspired spine. (c) Cross section of the artificial spine: sagittal view. (d) The arrangement of thirty two force sensors in the spine; cubic with red contour stands for the force sensor embedded in the silicon blocks represented by the cubic with black contour.

The dynamics of this spine is complex due to the flexibility and compliance of this spinal structure. As a result, non-linearity arises from the changing geometrical spine configuration during locomotion. Since the needed non-linearity can be outsourced to the physical body (compliant spine), adding a linear feedback and a linear readout are enough to form persistent memory and generate rhythmic movements. Therefore, this compliant spine is possible to be a computational source.
9.2.2 Information processing with spine dynamics

Fig. 9.2 shows an overview of the information processing in the proposed architecture. The robot’s dynamics are generated by the spine movement driven by four motors. The states of the spine measured through force sensors (FSR40) are sent to linear and static readout units that compute an output of the system \( O(t) \) as a sum of values of the force sensors \( FS_j(t) \) multiplied by output weights \( w_{\text{out}} \), where \( N \) is the number of force sensors:

\[
O(t) = \sum_{j=1}^{N} w_{\text{out},j} FS_j(t),
\]

where, \( w_{\text{out},j} \) and \( FS_j(t) \) indicate the output weight for \( j \)-th force sensor and the value of the sensor at time \( t \), respectively. The output weights \( w_{\text{out}} \) are updated so that the system reproduces a desired signal. The readout unit works as a pattern generator that computes motor command \( O(t) \) for the next time step.

For learning we collect the state \( FS_j(t) \) of force sensor \( j \) at every time step \( t = 1, 2, \cdots, M \) in a \( M \times N \) matrix, \( S \). The desired signal \( D(t) \) is also stored as a vector \( D = (d(1), d(2), \cdots, d(M)) \). Finally, the optimal output weights \( w_{\text{out}} \) are calculated by \( w_{\text{out}} = S^+ D \), where \( S^+ \) stands for the (Moore-Penrose) pseudo-inverse of \( S \).

This spine reservoir involves two nontrivial computational processes. First temporal integration of information, and second, the nonlinear combination of such temporally integrated information, which has the potential to be a computational source. We will demonstrate that the learning of complex, nonlinear dynamic process can be reduced, throughout the help of the physical body (morphology), to the much simpler task of learning some static, linear weights in the next section.
9.2.3 Teaching signals

We tested whether the proposed architecture can generate different behaviors corresponding to spinal movements, by collecting force measured from randomly distributed sensors around the spine structure.

Table 9.1: controllers for spine movement

<table>
<thead>
<tr>
<th>controller</th>
<th>((f_d, \phi_d))</th>
<th>((f_l, \phi_l))</th>
<th>((f_r, \phi_r))</th>
<th>((f_u, \phi_u))</th>
</tr>
</thead>
<tbody>
<tr>
<td>(SM_B)</td>
<td>((\frac{1}{\pi}, \pi))</td>
<td>*</td>
<td>*</td>
<td>((\frac{1}{\pi}, 0.0))</td>
</tr>
<tr>
<td>(SM_T)</td>
<td>*</td>
<td>((\frac{1}{\pi}, 0.0))</td>
<td>((\frac{1}{\pi}, \pi))</td>
<td>*</td>
</tr>
<tr>
<td>(SM_R)</td>
<td>((\frac{1}{\pi}, \pi))</td>
<td>((\frac{1}{\pi}, 0.0))</td>
<td>((\frac{1}{\pi}, \pi))</td>
<td>((\frac{1}{\pi}, 0.0))</td>
</tr>
</tbody>
</table>

The spine of Kitty robot is controlled with periodic motor commands given by Eq.9.1 using parameters shown in Table 9.1. Note that motors whose parameters are marked with asterisks in the table are controlled with constant values to keep the natural length without stretching and relaxing. The bounding gait, as a result of spinal flexion-extension movement, is generated by the controller \(SM_B\) which only controls the upside and downside strings located at the spine. In contrast, the trotting gait, generated by spinal lateral movement, can be achieved by taking the controller \(SM_T\), which only drives the side strings. Turning right behavior can be realized by the controller \(SM_R\), which combines \(SM_B\) and \(SM_T\) together.

Pattern generators \(O_i(t)\) for locomotive behaviors corresponding to spine movement \(SM_i(i \in \{B, T, R\})\) were designed so that they output periodic motor commands for the next time step and feedback this command to the spine reservoir.

9.2.4 Experiment procedure

To achieve gait generation, three phases are used: teaching, learning, and evaluating in these three behaviors. For each behavior, we conducted five trials each of which consists of 1,280 time steps. In the teaching phase, the teaching data to be used to train the reservoir readout is generated for 600 time steps, after initial 200 samples were discarded as transients. After training, the reservoir is implemented and the output weights are updated and fixed. Then it goes to evaluating phase for 480 time steps.
9.3 Results

9.3.1 Versatile behavior

Fig. 9.3, 9.5, 9.7 show the best performance generated by the spine reservoir in bounding gait, trotting gait and turning left behavior through the interaction with the real world. Here the environment is the carpet. The generated control signals are relatively periodic and similar in terms of the shape. Fig. 9.4, 9.6, 9.8 show that the frequency of target and desired signals are very close. These results indicate that this compliant spine can be regarded as a computational device to generate repetitive movements, in addition to be a mechanical component connecting the front legs and rear legs. Indeed, multiple behaviors can be produced by the same body, simply by changing the feedback weights.

Due to the lack of leg actuation, Kitty’s rear legs are sliding on the ground most of time. As a consequence, Kitty is more sensitive to the terrain. If the terrain is slightly uneven, Kitty gets stuck for a short moment, thus causing a time delay, compared to the target signal. Because of the size constraint of the arena, Kitty can’t get trained long enough with the aim to better emulate target signal. So our next step is to train Kitty by a treadmill.

9.3.2 Stability

In this section, the system was tested with different level of external perturbation. The number of evaluating phase is 480 time steps. In the first 160 time steps, the robot was moving without any load. In the second 160 time steps, a weight was added to the front body. Then the weight was taken away in the last 160 time steps.

Table 9.2 shows the behavior of robot with different external weight ranging from 150 g to 400g. We observed that the speed of Kitty decreases with the increase of the external weights. Fig. 9.9 is the result with the weight of 400g. We observed that after putting the weights at time step 160, the performance gets affected a little and recovered from this perturbation after time step 170. At time step 320 when the weight was taken away, the performance of Kitty was disturbed again. Then kitty regained its original pattern after time step 340.

9.4 Conclusion and Discussion

In this study, we demonstrated the developed compliant spinal structure featuring compliant elements is not only playing a passive role in signal pro-
Figure 9.3: The performance of pattern generator in evaluating phase: Bounding gait. Four subplots from top to bottom are the results of pattern generators for motor controlling the up, down, right, left side of the spine, respectively. Y-axis stands for the amplitude sent to the motor. X-axis indicates the time steps. The blue curve is the target trajectory and the red curve is the actual output from the spine reservoir.
Figure 9.4: Frequency analysis using Fourier transform: Bounding gait. Four subplots from top to bottom are the results of pattern generators for motor controlling the up, down, right, left side of the spine, respectively.

Figure 9.5: The performance of pattern generator in evaluating phase: Trotting gait. X and Y axes in four subplots represent the same meaning as Fig. 9.3.
Figure 9.6: Frequency analysis using Fourier transform: Trotting gait. X and Y axes in four subplots represent the same meaning as Fig. 9.3.

Figure 9.7: The performance of pattern generator in evaluating phase: Turning left. X and Y axes in four subplots represent the same meaning as Fig. 9.3.
Figure 9.8: Frequency analysis using Fourier transform: Turning left. X and Y axes in four subplots represent the same meaning as Fig. 9.3.

Figure 9.9: The performance of pattern generator in evaluating phase of bounding gait with external perturbation. The area marked in green is unstable state because of the external weight of 400 g.
Table 9.2: Performance of spine reservoir with different external perturbation

<table>
<thead>
<tr>
<th>external perturbation</th>
<th>No</th>
<th>150 g</th>
<th>300 g</th>
<th>400 g</th>
</tr>
</thead>
<tbody>
<tr>
<td>mean velocity</td>
<td>9.81 cm/s</td>
<td>9.78 cm/s</td>
<td>8.2 cm/s</td>
<td>7.6 cm/s</td>
</tr>
<tr>
<td>standard derivation</td>
<td>0.34</td>
<td>0.33</td>
<td>0.67</td>
<td>0.9</td>
</tr>
</tbody>
</table>

cessing and in achieving a set of desired behaviors, but rather serves as a computational source to achieve different behavior such as bounding gait, trotting gait, and turning left behavior. The results suggest that this computational source (compliant spine) is able to encode movement patterns, produce rhythmic patterns, learn new pattern. Interestingly, we found that multiple behaviors can be produced by the same physical body, simply by changing the feedback weights.

Moreover, we tested the stability of this spine reservoir by putting some external weights on the robot and found that this reservoir can resist external perturbation until 400 g, almost 30% of its own weight.

In all of the experiments, we found that the actual output can not precisely emulate the desired signal, due to the limited number of teaching data. But anyhow it shows the capability of learning of this spinal reservoir. For our future plan, we will train Kitty in a treadmill to get enough training data in order to emulate the desired signal better.
Bibliography


Francisco J Valero-Cuevas, Madhusudhan Venkadesan, and Emanuel Todorov. Structured Variability of Muscle Activations Supports

164


167


[206] Stefan Schaal, Jan Peters, Jun Nakanishi, and Auke Ijspeert. Learning movement primitives. In Paolo Dario and Raja Chatila, editors,


Appendix A

The mass and inertial matrix $M(q_e)$:

\[
M_{e11} = 2l_1a_1m_1 - l^2_1m - l^2_1m_1 - a^2_1m_1 - I_1 \\
M_{e12} = l_2a_1m_1\cos(q_{1e} - q_{2e}) - l_1l_2m_1\cos(q_{1e} - q_{2e}) \\
M_{e21} = l_2a_1m_1\cos(q_{1e} - q_{2e}) - l_1l_2m_1\cos(q_{1e} - q_{2e}) \\
M_{e22} = 2l_2a_2m_2 - l^2_2m - l^2_2m_1 - l^2_2m_2 - a^2_2m_2 - I_2
\]

The jacobian matrix $J(q_e)$:

\[
J_{e11} = l_1gm + l_1gm_1 - a_1gm_1 \\
J_{e12} = 0 \\
J_{e21} = 0 \\
J_{e22} = l_2gm + l_2gm_1 + l_2gm_2 - a_2gm_2
\]

The stiffness matrix $K$:

\[
K_{11} = -K_\theta \\
K_{12} = K_\theta \\
K_{21} = -K_\theta \\
K_{22} = K_\theta
\]