

Discrete and Rhythmic Motor Primitives for the Control of Humanoid Robots

THÈSE N° 4819 (2010)

PRÉSENTÉE LE 19 NOVEMBRE 2010

À LA FACULTÉ INFORMATIQUE ET COMMUNICATIONS

GROUPE IJSPEERT

PROGRAMME DOCTORAL EN INFORMATIQUE, COMMUNICATIONS ET INFORMATION

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

Sarah DÉGALLIER ROCHAT

acceptée sur proposition du jury:

Prof. R. Clavel, président du jury

Prof. A. Ijspeert, directeur de thèse

Dr R. Boulic, rapporteur

Prof. L. Sentis, rapporteur

Prof. D. Sternad, rapporteur



ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Suisse
2010

RÉSUMÉ

Le contrôle de robots avec beaucoup de degrés de libertés (DDLs) est un problème difficile, notamment parce que planifier des trajectoires multi-dimensionnelles complexes dans un environnement qui évolue avec le temps est un processus laborieux et lourd en termes de calculs nécessaires. Dans ce mémoire, nous proposons une architecture de contrôle où la planification des trajectoires (dans le sens de la définition de la tâche à accomplir) est découplée de la phase de génération de ces trajectoires, et ceci grâce à l'usage de primitives motrices, c'est-à-dire, de trajectoires dont la dynamique est prédéfinie avec certains paramètres contrôlables.

Le concept de primitives motrices est inspiré de l'étude du système moteur des vertébrés: les animaux sont non seulement capables de réaliser des tâches complexes de manière robuste, mais aussi de s'adapter rapidement aux changements de l'environnement. La planification du mouvement et la réelle génération des trajectoires est très probablement découplée chez les vertébrés: l'activation spatio-temporelle des muscles est produite au niveau de la colonne vertébrale par des réseaux de neurones appelés générateurs de patrons centraux. Ces réseaux sont activés par des commandes de contrôle simples, non-schématiques, et il semble donc que seuls les paramètres clés du mouvement doivent être produits par le cerveau pour générer un mouvement.

Nous développons ici une architecture de contrôle pour la génération de mouvements discrets et rythmiques basée sur des primitives motrices. Parmi les avantages de notre approche, soulignons que (i) la phase de planification est simplifiée grâce aux primitives motrices, dans le sens où les seules commandes de contrôle nécessaires se réduisent aux caractéristiques clés du mouvement, (ii) l'implémentation assure une transition lisse entre différents types de mouvements (discrets et rythmiques), (iii) la dynamique de la primitive motrice peut être modulée par l'information sensorielle pour des réponses adaptatives rapides et (iv) plusieurs DDLs peuvent être couplés ensemble pour garantir un comportement coordonné. En plus, la méthode a un coût calculatoire qui est bas et est de ce fait appropriée pour des applications qui nécessitent des boucles de contrôles rapides. Pour illustrer l'efficacité de l'architecture, nous l'appliquons à deux tâches: (a) la batterie interactive et (b) la marche à quatre pattes des enfants et le geste d'atteinte.

Mots clefs: Locomotion, geste d'atteinte, générateurs de patrons centraux, primitives motrices, systèmes dynamiques.

ABSTRACT

Controlling robots with multiple degrees of freedom (DOFs) is still an open and challenging issue, notably because planning complex, multidimensional trajectories in time-varying environments is a laborious and costly process. In this dissertation, we propose a control architecture where the planning consists in defining the key characteristics of the desired movement (e.g., the target for reaching), while the generation of the complete trajectory is based on predefined dynamics. More precisely, the generation of joint trajectories is decoupled from high-level planning (i.e., the definition of the task) through the use of a combination of discrete and rhythmic motor primitives, that is, movements with predefined dynamics.

The concept of motor primitives is inspired by the study of the motor system of vertebrates : animals are capable not only of performing highly complex tasks in a robust way but also of rapidly adapting to changes or uncertainties in the environment. Interestingly, the planning of movements and the actual generation of trajectories are most likely decoupled in vertebrates: the actual spatio-temporal sequence of activation of the muscles is produced at the spinal level through neural networks called central pattern generators (CPGs). These networks are activated by simple, non-patterned control signals from the brain, that is, only the key parameters of the movement seems to be needed from the brain for a task to be completed.

Here we develop a control architecture for the generation of both discrete and rhythmic movements based on motor primitives that has the following attributes (i) the planning phase is simplified thanks to the motor primitives, in the sense that the control commands that are required are reduced to the key characteristics of the movement, (ii) the implementation intrinsically ensures smooth transitions between different tasks (discrete and rhythmic) (iii) the dynamics of the motor primitives can be modulated by sensory feedback in order to have fast adaptive responses and (iv) several DOFs can be coupled together to ensure coordinated behaviors. In addition, this method has a low computational cost and is well-fitted for applications requiring fast control loops. We illustrate the efficiency of the architecture through two applications: (a) interactive drumming with the Hoap2 and the iCub and (b) infant-like crawling and reaching with the iCub.

Keywords: Locomotion, reaching, central pattern generators, motor primitives, dynamical systems.

REMERCIEMENTS

(ACKNOWLEDGEMENTS)

Ce manuscrit est le résultat d'un peu plus de 4 ans de recherche à l'EPFL. Beaucoup de gens m'ont accompagnée le long de ce chemin et j'aimerais les remercier ici. Ces remerciements sont parfois en anglais, parfois en français, selon la personne à laquelle ils sont adressés.

First of all, I would like to thank the members of the thesis committee, Dagmar Sternad, Ronan Boulic, Luis Sentis and Reymond Clavel, for their time and constructive comments that helped me to improve the final version of this document.

J'ai eu la chance de travailler dans un environnement extrêmement sympathique et je tiens à remercier le directeur de cette thèse, Auke, pour son éternelle bonne humeur, mais aussi pour sa confiance et pour toutes les opportunités qu'il m'a données. J'aimerais aussi remercier Marlyse pour sa disponibilité et sa gentillesse, ainsi que Sylvie pour le vent de fraîcheur qu'elle a amené dans le labo, pour son sourire et aussi pour les chocolats, gentille attention qui m'a profondément touchée!

Je ne peux que me sentir envahie par une vague de nostalgie en pensant au "bureau du bonheur", et je tiens à remercier ses occupants, Pierre-André et Ludovic. Pierre-André, tout d'abord merci pour "La minute de Monsieur C++", tes connaissances sans fin m'ont certainement épargné des heures de déboguage obscures et chaotiques, mais aussi, et surtout, merci pour ton amitié, pour les cafés "coup de gueule" et pour ton authenticité! Ludovic, merci pour la collaboration sur le crawling, j'ai appris beaucoup de choses et pu réutiliser beaucoup de code (malgré ta persistante grève contre les commentaires), mais surtout c'était vraiment très chouette de boire plein de vin et de bières en parlant mathématiques, philosophie et politique ! D'ailleurs je dois dire que je garde presque un bon souvenir de la pizzeria en bas du IIT et de son vino della casa.

Jérémie, merci pour les millions d'informations loufoques et décalées que ta curiosité débordante m'a permis de connaître! Merci pour ton amitié et ta compréhension, et merci de m'avoir écoutée et soutenue lorsque je n'allais pas bien! Alexander, thanks for being my twin PhD student! It was great to have the opportunity to share all the deadlines and other traumatic stuff with you. Going together to Vo Vietnam was also very cool, even though you broke my jaw once. I hope you will come back one day! Jonas, thanks for the comments on this dissertation, and also for all the "Pause" and the parties! Soha,

thanks for being THE other girl of the lab. I was very happy to get to know you better, and I hope we will have many occasions to see each other again! Jesse, thanks for saying p***, Dutch is a great language. Oh, and I am so proud of being a friend of the Guardian of the Internet! Andrej, thanks for organizing so many social events, for bringing some political debates into the lab and ... for living in such an awesome place! Sébastien, merci d'avoir accepté de me suivre dans ce temple de l'austérité qu'est le IIT. Merci de m'avoir aidée pour la démonstration finale et de t'être occupé de ce qui restait de moi après! Alessandro, merci d'avoir réparé toutes mes gaffes informatiques (à répétition parfois, maudit network manager!), mais aussi pour tous les jeux que l'on a organisés ensemble, les cadeaux de Noël et surtout les chasses au trésor! Yvan, merci pour ton aide avec Webots, j'espère que l'on se croquera bientôt sur les pistes de ski de fond de la Vallée! Mostafa, thanks for being a child at heart!

I would also like to express all my gratitude to the RobotCub team at IIT: Francesco Nori, Lorenzo Natale, Giorgio Metta, Matteo Fumagalli, Serena Ivaldi and Marco Ranzazzo. Francesco, thanks for your patience and for having (almost) always believed that we could do the crawling demo. Serena, thanks a lot for being so friendly, for the dinners at your place and for the pesto recipe!

J'ai la chance d'avoir de merveilleux amis et une famille extraordinaire, et j'aimerais les remercier ici. Il m'est difficile de résumer en quelques mots ma gratitude envers tous ces gens et j'espère qu'ils me comprendront au-delà des mots.

Merci à Isabelle et à Fanny pour leur amitié précieuse et pour l'organisation de mon enterrement de jeune fille ... Je n'aurais jamais pu imaginer les vertus anti-stress du Sumo avant! Merci à Myriam, ma partenaire-cousine, pour avoir cohabité avec moi au Great Escape pendant quelques temps... Merci à Alice pour les 4 bières, meilleur ami, bulbizarre et autres! Merci à Telma pour les cours de cuisine et à Anouck pour les cours de planche à voile! Merci à Jelena pour toutes les couleurs pétantes! Merci à Jacques pour les gchat endiablés à toute heure de la journée! Tack så mycket för din förståelse, kompis Anna!

Merci Michael et Gaby pour m'avoir soutenu tout au long de ma thèse, et aussi pour avoir créé ce petit être génial qu'est Evan, qui m'a souvent permis de relativiser les différentes pressions de la thèse.

Merci Paul, parce que, malgré ta vie difficile, tu as su rester heureux et te réjouir du bonheur des autres. Tu es un magnifique exemple de vie pour moi, et j'espère pouvoir vivre avec autant de sagesse que toi!

Papa, merci pour avoir toujours cru en moi et m'avoir toujours poussé à aller plus loin. Merci pour toutes les sorties, sportives ou autres (je ne parlerai pas du Château ici), et pour avoir toujours été à l'écoute. Maman, merci pour ton cœur plus gros que le monde entier, merci pour ton amour et ton soutien, merci d'avoir toujours été là quand j'en avais besoin.

Yannick, mon mari-superhéros, merci pour toutes les fois où tu as volé à mon secours, et merci parce que la vie est un pur bonheur à tes côtés.

CONTENTS

Résumé	i
Abstract	i
Remerciements	iv
1 Introduction	1
2 Context of the research	5
2.1 Dynamical systems and robotics	5
2.2 Legged locomotion	7
2.2.1 Model-based approaches	7
2.2.2 CPGs-based approaches	8
2.3 Discrete movements and reaching behaviors	10
2.3.1 Model-based approaches	11
2.3.2 Primitive-based approaches	12
2.4 Concluding remarks	13
3 Modeling discrete and rhythmic motor primitives	15
3.1 Introduction	16
3.2 A simplified view on motor systems	17
3.3 Defining discrete and rhythmic movements	18
3.4 The combination of discrete and rhythmic movements	21
3.5 Generation of discrete and rhythmic movements	22
3.5.1 Central pattern generators	22
3.5.2 Motor primitives and forces fields	25
3.6 Planning of discrete and rhythmic movements	27
3.6.1 Motor primitives in movement planning	27
3.6.2 Movement encoding by the motor cortex	28
3.7 Mathematical models	29
3.7.1 Two/Two hypothesis	30
3.7.2 One/Two hypothesis	35

3.7.3	One/One hypothesis	39
3.7.4	Two/One hypothesis	42
3.7.5	Discussion of the models	44
3.8	Concluding remarks	47
4	Control architecture	49
4.1	Introduction	50
4.2	Discrete System	51
4.2.1	Stability and analytical solution	52
4.2.2	Some Interesting Properties	52
4.3	Rhythmic system	55
4.3.1	Stability and analytical solution	55
4.3.2	Some Interesting Properties	56
4.4	Unit pattern generator	59
4.5	Central pattern generator	61
4.5.1	Some Interesting Properties	62
4.6	Concluding remarks on the control architecture	63
5	Application to Drumming	65
5.1	First Implementation on the HOAP-2	66
5.1.1	CPG network and parameters	66
5.1.2	Task definition and constraints	69
5.1.3	Score playing with the HOAP-2	70
5.1.4	Drum pads position feedback	72
5.1.5	Concluding remarks on the first implementation	73
5.2	Second Implementation on the iCub	74
5.2.1	Design of the whole-body CPG	75
5.2.2	Task specification and Constraints	76
5.2.3	Interactive drumming with the iCub	78
5.2.4	Contact Feedback	79
5.2.5	Visual feedback	82
5.2.6	Concluding remarks on the second application	84
6	Application to Crawling	85
6.1	Software Implementation	86
6.2	CPG design	87
6.3	Choice of parameters	90
6.4	Switching between crawling and reaching	91
6.5	Contact feedback (Work by Ludovic Righetti)	94
6.6	Combination with a high-level path planning	96
6.7	Concluding remarks	97

Contents

7 Outlook	99
7.1 Contributions	101
A Derivation of the radius of Eq. 4.8, Chapter 4	105
B Information on the robots	107
B.1 HOAP-2	107
B.2 iCub	107
B.2.1 Hardware	108
B.2.2 iCub Software	108
B.3 Webots	109
C List of contributors	111
Bibliography	111
Curriculum Vitae	123

CHAPTER 1

INTRODUCTION

Goals

This dissertation presents the development of an architecture for the generation of trajectories that is : (i) suitable for robots with many degrees of freedom and (ii) general enough to be applied to a wide range of behaviors. More precisely, the idea is to develop a system that can generate various trajectories without being task specific. In order to do so, movement generation in humans and animals is first studied to gain an insight into suitable solutions for robotics. A model based on the biological concept of motor primitives, i.e. template of movements that can be superimposed and concatenated, is then developed. The motor primitives are modeled by dynamical systems with global attractors that provides them with resistance against perturbations. By defining two types of motor primitives, corresponding to discrete and rhythmic movements, we will show that a repertoire of trajectories large enough to be applicable to various robotic applications can be obtained. The generality of the approach and its suitability for robots with many degrees of freedom is studied through its application to interactive drumming, crawling and reaching with the humanoid robot iCub.

Motivations

Planning trajectories for robots with many *degrees of freedom* (DOFs) is often addressed as follows : a model is built to derive the input commands that are required to obtain the desired movements. When a system is redundant (i.e. the number of constraints is smaller than the number of DOFs), the desired trajectory is obtained through an optimization process. However, in humanoid robots, the high redundancy and the large workspace of the system makes the search for a suitable plan challenging. Additional difficulties include time-varying environmental constraints and, for some applications such as bipedal walking, a limited time to find adequate solutions. Indeed, if for instance a perturbation occurs during walking, the computation of the new plan must be very fast

to avoid that the robot falls.

Looking at studies on movement generation in vertebrates provides an alternative approach to trajectory planning. Indeed, it seems that, in animals, the planning of the task to be performed is decoupled from the actual trajectory generation (i.e., the temporal sequence of activation of the muscles). More precisely, neurally encoded movements – *motor primitives* – have been brought to light in the spinal cord in various animals (Bizzi et al (2008); Grillner (2006)), indicating that (high-level) trajectory planning in animals could come down to the selection and the appropriate activation of pre-existing motor primitives. These primitives simplify the control of movements in the sense that the CNS only needs to activate a group of muscles – a *synergy* – instead of activating all the muscles involved in the movement individually.

This second approach provides an interesting perspective to traditional control methods. Fig. 1.1 shows a diagram illustrating the contribution of motor primitives. In the traditional approach, there are usually two different processes: a high-level planner that computes the desired trajectories and a low-level controller (e.g., a PID controller) that transforms the desired trajectories into motor commands, generally based on the error between the current states and the desired ones. The idea behind the concept of motor primitives is to add a middle-level controller to the system that is composed of a set of trajectories with predefined dynamics. In terms of robotic control, the motor primitives can thus be seen as sophisticated low-level controllers, in which a priori knowledge about the movements to be performed is embedded and that can be modulated according to feedback information. The advantage of using these primitives is threefold. First, they ease the planning problem by reducing the workspace of the robot through the constraint of the dynamics of the trajectories. Second, they provide the system with a fast, low-level feedback loop that can be used to rapidly correct trajectories according to the incoming sensory information if required (without the need for a new motor plan). Finally, different degrees of freedom can be coupled together to ensure inherent synchronization and coordinated behavior. In other terms, motor primitives provide an effective, dynamic way to embed a priori knowledge about the task into the low-level control system, as, for instance, arm synchronization for bi-manual tasks or trajectories with bell-shaped velocity profile for reaching movements. They thus provide a fundamental tool to develop efficient, fast architectures for the generation of movements, particularly in the case of robots with many degrees of freedom and meant to evolve in time-varying environments, such as humanoids.

However, the concept of motor primitives has been mainly used in robotics to model biologically plausible behaviors, with a focus on purely discrete or purely rhythmic tasks. In this dissertation, our goal is to develop a control architecture for trajectory generation for both discrete and rhythmic tasks that is efficient for robotic applications in general. In order to do this, we define the following specifications:

- motor primitives corresponding to both discrete and rhythmic movements will be

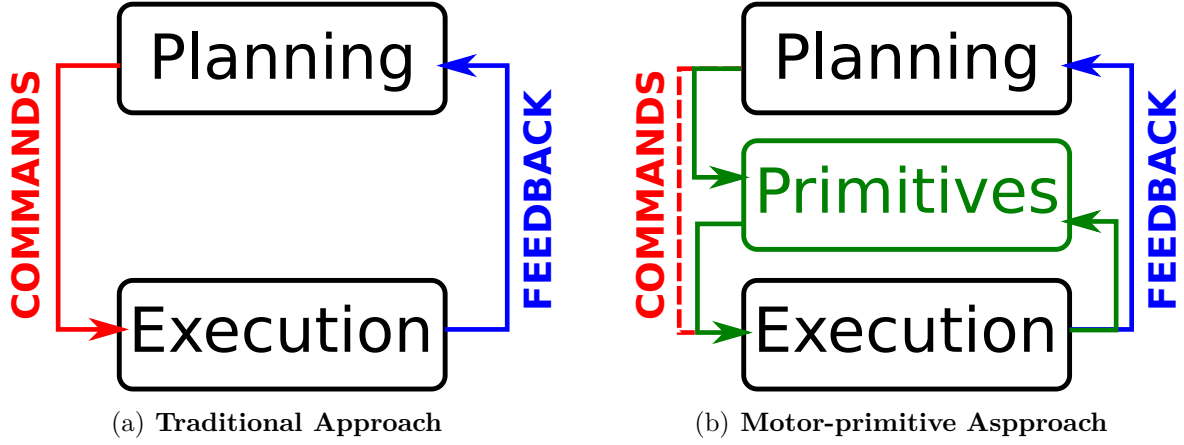


Figure 1.1: Motor primitives for control. (a) In the traditional approach, a planner computes the motor commands needed to achieve the task according to the feedback information. In redundant systems, the desired trajectory is usually found using optimization given a certain performance criterion. When the environment changes, inducing a modification of the feedback signal, the motor commands need to be computed again. (b) Motor primitives, that is, trajectories with predefined characteristics are added to the system. The generation of the commands is now divided into two steps: the definition of the open parameters of the motor primitives and the generation of the trajectories by the motor primitives. In addition to the main feedback loop, a low-level feedback loop for rapid modulations of trajectories can be added.

defined in order to ensure that a wide range of tasks can be completed properly;

- the primitives will be modeled by dynamical systems with goal attractors that enable the modulation of the trajectories according to the constraints or perturbations;
- the control parameters of the motor primitives will be explicitly linked to the characteristics of the outcome trajectory to make the architecture easy to combine with high-level planners.

Thanks to these motor primitives, we will develop an architecture for generating trajectories applicable to various tasks, as will be illustrated here through its application to drumming and crawling.

Main Contributions and Outline

The content of this dissertation can be divided into three main parts: (i) a study on the generation of discrete and rhythmic movements in humans and animals and its modeling

through a review of the existing literature; (ii) the development of a control architecture for discrete and rhythmic movements based on the application of dynamical systems; and (iii) the application of this architecture on a humanoid robot to accomplish two tasks: (a) drumming and (b) crawling and reaching. The main contributions of this work are

- An extensive review on the generation of discrete and rhythmic movements in vertebrates in the context of motor primitives;
- A classification of four possible types of models for the generation of discrete and rhythmic movements;
- A control architecture based on motor primitives that allows for
 - the generation of both discrete and rhythmic movements
 - the switch between these two types of movements
 - the integration of feedback modulations;
- The implementation of interactive, closed-loop drumming on the iCub robot, as well as a previous implementation on the HOAP-2 robot;
- The implementation of autonomous, closed-loop crawling and reaching on the iCub robot.

We start by a brief overview of some relevant, state of the art approaches to robotic control, with a focus on locomotion and reaching (Chapter 2). Then, in Chapter 3, we discuss in more details the biological concepts that inspired this research and we give an insight into the generation of discrete and rhythmic movements based on these concepts. We also briefly present existing models for the generation of both discrete and rhythmic movements. Chapter 4 presents the control architecture that we have developed for the generation of discrete and rhythmic movement. To demonstrate the adequacy of the architecture to robotic control we then present two applications: interactive drumming (Chapter 5) and crawling and reaching (Chapter 6). We then conclude by a discussion on the advantages and limitations of the approach, along with the future work.

A list of movies related to this thesis can be found at http://biorob2.epfl.ch/sd_movies, along with the Matlab code for generating most of the figures of this dissertation, as will be referenced in the pertaining chapters.

CHAPTER 2

CONTEXT OF THE RESEARCH

The aim of this research is to develop an architecture for the control of robots with multiple degrees of freedom (DOFs) for both discrete and rhythmic tasks. This problem is still open and challenging, notably because planning complex, multidimensional trajectories in time-varying environments is a laborious and costly process. In this work, we propose a control architecture where the generation of joint trajectories is decoupled from high-level planning of the task. This is done through the combination of discrete and rhythmic motor primitives, that is, trajectories which dynamics are predefined but with open parameters for variability. To model these primitives, we will rely on the theory of dynamical systems. Indeed systems with adequate attractor properties will result in motor primitives that are resistant against perturbations and parameter changes.

Before presenting our work, we present some of the relevant literature and how it relates to our work. We start with a brief review on the applications of dynamical systems to robotics. We then discuss control strategies for rhythmic and discrete tasks separately building on previous work where only one type of movement is usually considered. We will see that the concept of motor primitives provides an alternative to more traditional, model-based approaches.

Note that we do not present literature on the biological grounding of motor primitives here, as it will be extensively discussed in the next chapter.

2.1 Dynamical systems and robotics

The control of robotic devices using motor primitives modeled by dynamical systems has often been addressed in the literature, with applications to learning by demonstration – the so-called dynamical motor primitives (DMP) – [e.g., Ijspeert et al (2003), Gribovskaya and Billard (2008), Pastor et al (2009), Kober and Peters (2010)], rehabilitation (Ronsse et al (2010)) and locomotion [e.g., Maufroy et al (2008), Kimura et al (2007)]. Applications similar to our work, i.e. the generation of trajectories given simple, explicit high-level commands, can be found in Crespi et al (2008) and Maufroy et al

(2008) for locomotion and Bullock and Grossberg (1988) and Hersch et al (2008) for reaching. The novelty of our work is that we address the generation of both discrete and rhythmic movements through the same process, a subject that has received little attention so far (Degallier and Ijspeert (2010)). The existing models that we know, namely those developed by De Rugy and Sternad (2003), Schaal et al (2000), targeted at reproducing observations made in humans and have not been applied to robotic control. In a third model by Schöner and Santos (2001), discrete movements are generated by truncating rhythmic movements. These three models will be presented in detail in Chapter 3, Section 3.7.

In our case, a motor primitive that can generate both discrete (goal-directed) movements and rhythmic movements was developed. A motor primitive controls one DOF of the robot, and the motor primitives can be coupled together to coordinate several DOFs. The primitive will have three open parameters, that correspond to the main characteristics of the two types of movements: the frequency and the amplitude for rhythmic movements and the target for discrete ones (note that the velocity could also have been included).

To model this primitive, we will use dynamical systems as they can be designed to have interesting attractor properties which makes them well-suited for trajectory generation. Indeed, in some systems, the dynamics can be such that trajectories starting from any point in a given neighborhood in the space will eventually converge to a given position that is called an attractive fixed point. In other situations, the system can converge to a limit cycle, that is an attractive closed orbit in the state space. A limit cycle system is also called a (limit cycle) oscillator¹. Attraction, and in particular if it is global, is interesting for several reasons. Firstly, it gives the possibility to change parameters on the fly (e.g. to change the location of fixed point, the amplitude or the frequency of the limit cycle online), because it is known that the system will eventually converge to the newly defined attractor. Secondly, fixed points and limit cycles have an intrinsic robustness against small perturbations; if a perturbation occurs, the trajectory will resume to the attractive solution once the perturbation vanishes. Another interesting property of dynamical systems is that the whole dynamics can be modulated by introducing external signals, such as an external force or by adding a new attractor in the system. This latter property can be used to integrate sensory feedback terms. When the external force is rhythmic and under certain conditions, entrainment can occur, i.e. the overall frequency of the system synchronizes to the one of the external signal. This can be used to couple limit cycle systems together to control the phase shift between them. All these properties will be further illustrated in Chapter 4.

¹An oscillator is generally a set of differential equations that has a closed orbit as an equilibrium solution. It is called a limit cycle oscillator if the orbit is attractive and a harmonic oscillator otherwise (Strogatz (2001)). In this dissertation, the term oscillator will always refer to limit cycle oscillators.

2.2 Legged locomotion

Designing robotic controllers for legged locomotion is a very challenging area of research, balance control being one of the major issues. In addition, while efficient controllers for following predefined trajectories have been designed, adaptation to unpredicted variations of the outside world is still much unexplored.

Two main approaches to locomotion can be defined : model-based approaches for slow locomotion on challenging terrain and CPG-based approaches for fast locomotion with possible change in gaits. We briefly present these two approaches, with a focus on quadruped locomotion as we are interested in crawling.

2.2.1 Model-based approaches

Key to legged locomotion is the control of balance. Several criteria have been developed to characterize stability, as for instance the zero moment point (ZMP), the central moment pivot (CMP) or the Foot Rotation Indicator (FRI) [see Popovic et al (2005) for related references]. Note that all these criteria provide sufficient but not necessary conditions and thus constrain the space of possible stable walking trajectories. They do not provide by themselves solutions to the stability problem.

The criterion the most commonly used in literature is the ZMP, which is defined as *the point of resulting reaction forces at the contact surface between the extremity and the ground* (Vukobratovic and Juricic (1969)). To ensure stability, this point should never leave the convex hull defined by the contact points. By definition, the ZMP cannot leave the hull, which implies that the situation where the ZMP lies at the border of the hull is undetermined (i.e. we cannot know whether the robot is stable or not). In addition the ZMP criterion hold only in cases where the contact points are in the same plan. Impressive results based on the ZMP include the work with the Asimo robot by Honda (Fig 2.1(a)).



(a) Asimo



(b) Little Dog

Figure 2.1: Model-based controlled robots. On the left, the Asimo robot by Honda. On the right, the Little Dog robot developed by Boston Dynamics for the DARPA challenge.

The DARPA Learning Locomotion project is a good illustration of a model-based approach to locomotion: Several institutions were challenged to develop a robust controller for locomotion over rough terrain. The Little Dog, illustrated in Fig. 2.1(b), is a robot developed by Boston Dynamics, Inc, specially for that project. Knowledge of the terrain is assumed, but the same controller has to be used for various terrains. The results were evaluated by the speed with which the robot was able to cross the terrain. In most of the approaches [e.g. Zico Kolter and Ng (2009), Kalakrishnan et al (2010) or Zucker et al (2010)], a high level planning algorithm is used to find possible foot placements using expert knowledge. The trajectories of the feet in the Cartesian space are then found using splines (i.e. piecewise-defined polynomials) that are computed in order to avoid collision constraints. Thanks to this technique, impressive results can be obtained, with the robot being able to clear obstacles as big as the size of its leg length.

Note that a very interesting, intuitive approach for stability and locomotion which does not require dealing with a complex model of the robot is the Virtual Model Control derived by Pratt et al (2001) and his colleagues at the Leg Laboratory in MIT. The idea is to introduce virtual components inducing virtual forces instead of dealing with the complex, nonlinear dynamics of the robot. In Pratt et al (2001) a "virtual granny walker mechanism", with a spring-damper system, is introduced to maintain the pitch to zero and the height of the robot to a constant value to ensure stability, while a "virtual dog track bunny mechanism", applying a virtual force in the direction of the movement, assures that robot velocity is constant.

2.2.2 CPGs-based approaches

In biology, a fundamental notion for the generation of rhythmic movements is central pattern generators (CPGs) [see Grillner (2006) for a review]. CPGs are neural circuits, which are able to produce rhythmic signals as outputs with simple (non-rhythmic) inputs and responsible for the generating various behaviors in vertebrates (Delcomyn (1980)), notably locomotion (Grillner (1985)). This concept is interesting for modeling locomotion because it simplifies the high-level control, but also because the gait can be intrinsically defined in the network, as will be further explained in Chapter 4. In addition entrainment can be used to couple the controller to the robotic system.

CPGs-based approaches are often used as a tool to better understand biology. For instance, in our lab, a biologically inspired amphibious snake (or eel/lamprey-like), a salamander, a fish and a centipede-like robots have been built to test hypotheses about animal locomotion, but also to develop efficient controllers for robotic locomotion. Thanks to the Salamandra Robotica (Fig. 2.2(b)), a salamander-like robot, possible explanations for the change of speed and directions of the salamander were provided, as well as a model for the switch between walking and swimming (Ijspeert et al (2007)). An extensive review of the CPG-driven robots can be found in Ijspeert (2008).

In a seminal article, Taga (1994) presented a controller for biped locomotion where

2.2. Legged locomotion

a stable gait emerged through global entrainment between the rhythmic activity of the neural system, the movements of the musculo-skeletal system and the interaction with the changing environment. The model was applied in simulation to a bipedal planar robot, and it was shown that the controller was intrinsically robust against large perturbations. This result was extremely promising, although a lot of fine tuning was required even for this simplified case.

Other interesting approaches to locomotion include the work of Morimoto et al (2008) and of Aoi and Tsuchiya (2007). Morimoto et al (2008) developed a controller for locomotion based on CPG and a simple model of the COP as an inverted pendulum. The controller was successfully applied to two robots, the CB and a small humanoid robot. As for Aoi and Tsuchiya (2007), they developed a controller for the HOAP-2 where the CPG is adapted according touch sensing information to obtain a turning behavior.

Quadruped locomotion has been extensively studied by Kimura and his colleagues. They have developed several robots (Collie, Patrush, Tekken (2.2(a)), Kotetsu) in which the mechanics and the controller where designed in parallel. Their control architecture closely reproduces observations made on mammals (Kimura et al (2007)). The control of balance is ensured through a set of reflexes and responses (where a reflex acts directly on the output trajectory while a response modulates the CPGs), which allows the robot to move very fast on terrains with different frictions, slopes and to clear different types of obstacles.



(a) Tekken 4



(b) Salamandra Robotica

Figure 2.2: CPGs-controlled robots. On the left, Tekken 4, the robot developed by Kimura’s lab to study closed-loop quadruped locomotion. On the right, Salamandra Robotica, the robot developed by Ijspeert’s lab to better understand the control of swimming and locomotion in the salamander.

Both the model-based approach and the CPG-based approach have their pros and cons: on the one hand, there is no clear methodology on how to design a CPG, while methods exist to design ZMP-based trajectories, and on the other hand, it is difficult to modulate the policies obtained according to the ZMP fast enough, e.g to deal with model mismatch.

When thinking about the different types of terrain that a robot may face, or when thinking about locomotion in vertebrates, it seems that both approaches would be needed: e.g. a CPGs-based approach for running and walking on normal terrains and a model-based approach for challenging situations that requires precise foot placement, such as icy surfaces or rough terrains. It would thus be interesting to have an architecture that can offer both types of control and an architecture that can generate both discrete and rhythmic trajectories thus bridging the gap between the two approaches. Indeed one could use rhythmic primitives for natural locomotion, with fast speed and changes of gaits, and discrete primitives for challenging terrains (in the same way that splines were used in the approaches that we have presented) .

2.3 Discrete movements and reaching behaviors

Concerning the generation of discrete movements, most approaches in robotics rely on the fact that many invariants in the generation of reaching movements have been uncovered by neuroscientists during the last decades [see Gibet et al (2004) for a review]. Among them, we cite the following:

- Invariance of the velocity profile: It has been shown that the global shape of the velocity profile for reaching movements is approximatively bell-shaped, with an asymmetry that depends on the speed of movement (Morasso (1981)).
- Isochrony Principle and Fitts' Law: It has been shown that the dependence of the duration of the execution of the movement on its amplitude is negligible. Fitts' Law quantifies this constancy:

$$T = a + b \log_2 \left(\frac{2A}{W} \right) \text{ or } T = a + b \log_2 \left(\frac{A}{W} + c \right)$$

with A the amplitude of the movement, W the width of the target, a , b constants determined in an empirical way and $c = 1$ or $c = 1/2$ depending on the model.

- Two-Third Power Law: It has been shown that the angular velocity for drawing elliptical curves can be related to the curvature of the trajectory. More precisely, the so-called Two-Third Power law is given by:

$$\omega(t) = kC(t)^{2/3}$$

where ω is the angular velocity, C the curvature of the end effector trajectory and k is a constant.

- Smoothness of the movement: The trajectories that we generate minimize the jerk (i.e. the derivative of the acceleration) to attain a certain smoothness (Minimum Jerk Model).

2.3. Discrete movements and reaching behaviors

To explain these invariants, two main categories of answers can be found in the literature. The first one supposes that the brain explicitly computes a desired trajectory based on an internal model of the whole body system, this trajectory matching some optimality criterion. In the second one, these properties are believed to emerge from the interaction of sensory, muscular and neural systems (Bullock and Grossberg (1988)). These two approaches are thus very similar to the ones that we defined for locomotion, and we will refer to them as model-based and primitive-based approaches.

2.3.1 Model-based approaches

An internal model is defined as *a system which mimics the behavior of a natural system*. Two kinds of models can be defined, namely forward internal models and inverse internal models (Wolpert and Kawato (1998)). A forward model predicts the next state of the body given its current position and the motor command that are sent. and conversely, an inverse model gives the motor commands that will produce the desired next state of the body. Note that, while Shadmehr and Mussa-Ivaldi (1994) provided convincing evidence that the CNS is using inverse models through a manipulandum experiment², their existence and also the need for them is still debated [e.g. Bridgeman (2007) or Feldman (2009)].

Inverse models provide tools that are well suited for control, as they give the command needed to attain a desired state. As for the forward models, they provide an estimation of the current state (they are used to cope with the fact that sensory information about the actual state is time-delayed). Wolpert and Kawato (1998) proposed a model where multiple inverse and forward models are used to control movements depending on the task and the state.

An issue central to these models is the so-called Bernstein problem (Bernstein (1967)), namely redundancy. Indeed the motor command corresponding to a given movement is not unique, as the number of degrees of freedom is largely superior to the constraints. A way to handle this redundancy is to define an optimality criterion according to which the trajectory will be selected. The theory of optimal control is well-suited to handle this problem, as it consists on finding the command that generates the optimal trajectory. Optimality criteria are usually derived from the invariants of movements mentioned ear-

²In this experiment, a user has to move the handle of a manipulandum to a given target position. The target and the position of the handle is shown on a monitor, but the user does not see his/her own arm. The manipulandum has two actuators at its basis to produce desired torque; a torque is applied so to produce a force field depending on the velocity of the hand. When no force field is applied (case A), the user achieves to reach the target, and the trajectory is straight. When the force field is applied (case B), the user makes an error and needs to correct the trajectory to reach the target. As the number of trials increases, the error decreases and the trajectories are straight again, possibly indicating the existence of a kinematic plan. When the force field is removed (case C), the errors obtained are approximatively the mirror images of the previous ones, indicating that the model of the dynamics that was learned in case B is still used by CNS.

lier, as, for instance, the minimum jerk, the minimum variance or the minimum torque change [see Tran (2009) for a detailed review]. The major drawback of optimal methods being that it is then difficult to apply in real-time due to the large size of the search space.

To overcome this problem, Todorov et al (2005) proposed a model inspired on motor primitives, where a low-level controller that is responsible for the generation of the trajectory is coupled to a high-level controller that consider only task parameters. In other words, errors are corrected only if they are relevant to the task. Such a method drastically reduces the search space thus enabling the control a 3D model of the arm composed of 7 DOFs and 14 muscles (Liu and Todorov (2009))

2.3.2 Primitive-based approaches

In the second approach, it is postulated that the invariants emerge from the dynamics of the body. Bullock and Grossberg (1988) developed a model based on dynamical systems, called the vector-integration-to-endpoint (VITE), where the motor commands are only the target position and a function that gates the onset of the movement (the go command). Other characteristics of the trajectory simply emerge from the dynamics of the system. The VITE model can reproduce many of the invariants and observations made on humans³. A simpler version of this system will be presented in Chapter 3, Section 3.7

The VITE model was later extended by Hersch et al (2008) to design a reaching controller based on multi-referential dynamical systems. Two concurrent dynamical systems, one in task space and the other in joint space are used to define the movement, coherence being ensured through a constraint. This approach avoids singular configurations and provides an intrinsic solution to the issue of joint angle limitations.

Another interesting approach to reaching is the one proposed by the dynamic motor primitives (DMPs). The approach was originally developed by Ijspeert et al (2002) to learn both discrete and rhythmic movements through demonstration. This is done through the learning of landscapes of point attractors to form control policies. The essence of this approach is in anchoring Gaussian basis functions in the dynamical system, the weights of the basis functions being learned through nonlinear regression techniques. Once a reaching movement is learned, the dynamics can be adapted for new targets, but also to deal with static and dynamic obstacles along the trajectory

³More precisely, *quantitative simulations are provided of Woodworth's law, of the speed-accuracy trade-off known as Fitts's law, of isotonic arm-movement properties before and after deafferentation, of synchronous and compensatory "central-error-correction" properties of isometric contractions, of velocity amplification during target switching, of velocity profile invariance and asymmetry, of the changes in velocity profile asymmetry at higher movement speeds, of the automatic compensation for staggered onset times of synergetic muscles, of vector cell properties in precentral motor cortex, of the inverse relation between movement duration and peak velocity, and of peak acceleration as a function of movement amplitude and duration.* (Bullock and Grossberg (1988))

2.4. Concluding remarks

(Pastor et al (2009), Stulp et al (2009)).

Recently, Polyakov et al (2009) proposed a model that brings optimization-based and primitive-based approaches together. The idea is to perform the optimization process on a space of solutions defined by motor primitives and their combinations. For the primitives, they have chosen parabolic curves, for two reasons. First, this type of curves minimize the hand jerk and thus ensures that the movement is smooth. Second, they are invariant under the same type of transformations that the Two Thirds power law, that is, equi-affine transformations (i.e., transformations that preserve areas and parallelism). This latter condition is imposed so that a unique curve template, that can be affinely transformed and concatenated, can be used to obtain complex movements. Evidence of the existence of such parabolic motor primitive in monkeys during scribbling are also provided in the paper. In the field of virtual reality, Treuille et al (2007) showed that by using basis functions to represent the value space, they were able to obtain a fast, near optimal control of human characters for complex tasks.

2.4 Concluding remarks

Model-based approaches and CPG-based approaches used to be two distinct answers to movements generation in robotics. However, it seems to us that successful results could be obtained by combining them. Indeed, by decoupling the planning of the main characteristics of the movement and the actual generation of the trajectories, one has the opportunity to benefit from the powers of both approaches. In particular, modern control techniques for dealing with multiple constraints, such as the one presented in Sentis and Khatib (2005), can be a powerful complement to the biologically inspired concept of motor primitives.

In the next chapter, we present more in detail evidences for the existence of motor primitives in vertebrates. We discuss the modeling of discrete and rhythmic movement based on the concepts of CPG and force fields and we propose a classification of the different possible categories of models.

CHAPTER 3

MODELING DISCRETE AND RHYTHMIC MOTOR PRIMITIVES

In this chapter, we study the open issue of the generation of discrete and rhythmic movements in humans and in vertebrates in general based on the existing literature. We use the framework of motor primitives to try to gain insights on the motor structures that could be used in vertebrates to generate these movements.

Interestingly, rhythmic and discrete movements are frequently considered separately in motor control, probably because different techniques are commonly used to study and model them. Yet, an increasing interest for a comprehensive model for movement generation requires to bridge the different perspectives arising from the study of those two types of movements. In this chapter, we consider discrete and rhythmic movement within the framework of motor primitives, i.e. of modular generation of movements. Thereby we hope to get an insight into the functional relationships between discrete and rhythmic movements and thus into a suitable representation for both of them.

Within the framework of motor primitives, we can define four possible categories of modeling for discrete and rhythmic movements depending on the required command signals and on the spinal processes involved in the generation of the movements. These categories are first discussed relatively to biological concepts such as force fields and central pattern generators and are then illustrated by several mathematical models based on dynamical system theory. A discussion on the plausibility of these models concludes this chapter.

This chapter is organized as follows. After a brief introduction, we present a simplified model of the motor system on which we will base our reflection (Section 3.2). We then present several studies on the differences between discrete and rhythmic movements (Section 3.3) and some of the literature on the combination of these movements (Section 3.4). Although we are well aware that movement generation is a dynamic process involving the whole motor system, we discuss movement execution and movement planning separately, in Sections 3.5 and 3.6 respectively, since we think that in this way distinct

properties pertaining to those two phases of movement can be emphasized,. Finally we present in Section 3.7 some existing mathematical models for the generation of discrete and rhythmic movement, since such models might provide important information on the generation of these movements.

This chapter is a reproduction of the work presented in Degallier and Ijspeert (2010). The final publication is available at www.springerlink.com. We have omitted the section presenting our model, since it will be presented in details in the next chapter.

3.1 Introduction

Humans are able to adapt their movements to almost any new situation in a very robust, seemingly effortless way. To explain both adaptivity and robustness, a very promising perspective is the modular approach to movement generation: movements result from combinations of a finite set of stable motor primitives organized at the spinal level (see Bizzi et al (2008) for a review). In this chapter, a motor primitive is a network of spinal neurons that activates a set of muscles (that we call a synergy) in a coordinated way in order to execute a specific movement. Motor primitives are thus defined relative to the movement they produce.

In terms of control, the modularity assumption is attractive because it drastically reduces the dimensionality of the problem: instead of a complex stimulation of a vast number of muscles across the body, high-level commands can be summed up as activation signals for a finite, discrete set of motor primitives. Strong evidence, notably through the concepts of central pattern generators and force fields [see reviews by Grillner (2006) and Bizzi et al (2008)], supports the existence of such functional modules at the spinal level in vertebrate animals. For instance, Kargo and Giszter (2000) have demonstrated how a finite set of spinal motor primitives could account for the natural wiping reflex in the frog, showing that the central nervous system (CNS) could use such primitives to produce natural behaviors.

Assuming the existence of such motor primitives provides an interesting framework for reflecting upon the potential differences between discrete and rhythmic movements. It allows us to reflect on these movements relative to a simplified view of movement generation: a high-level command activates a (set of) motor primitive(s) at the spinal level that generates a given kinematic outcome. Given this scheme, we can consider the potential differences between discrete and rhythmic movements that are not related to sensory feedback or muscle interaction but to the spinal processes underlying them and to the high-level commands needed to activate these spinal processes. We call this approach a functional approach to distinguish it from the many studies focusing on the kinematics of these types of movements, such as, for instance, the thorough analysis by

3.2. A simplified view on motor systems

Hogan and Sternad (2007).

Most of the studies on discrete and rhythmic movements are either based on electromyographic (EMG) analyses of the generated movements (Hogan and Sternad (2007), van Mourik and Beek (2004)) or on functional magnetic resonance imaging (fMRI) analysis (Schaal et al (2004)) as will be reviewed in Section 3.3. While those studies have provided insightful results on the nature of discrete and rhythmic movements, we think that adopting a functional perspective is a useful, complementary step towards understanding the differences between the movements regarding the way they are generated, and also to gain more understanding on how brain and EMG studies can be bridged. Moreover, the generation of discrete and rhythmic movements at the spinal level has been extensively studied in vertebrates through the concepts of force fields and CPGs respectively, providing an interesting basis for reflection.

3.2 A simplified view on motor systems

In this section, we briefly present a simple model for movement generation based on the concept of motor primitives. We consider the processes underlying the generation of both movements with an emphasis on the contribution of the spinal component of the CNS. Such a simplified structure will provide us with a framework for discussion throughout this chapter.

According to textbooks [see, e.g., Kandel et al (2000)], movement generation is achieved through three motor structures organized hierarchically and corresponding to different levels of abstraction. These structures are (a) the *cerebral cortex*, which is responsible for defining the motor task; (b) the *brain stem*, which elaborates the motor plan to execute the motor task; and (c) the *spinal cord*, which generates the spatio-temporal sequence of muscle activation to execute the task. In addition, the cerebral cortex and the brain stem are influenced by the *cerebellum* and the *basal ganglia*, which can be considered as feedback circuits, the cerebellum being connected to the spinal cord as well.

In order to consider the relationships between discrete and rhythmic movements, we will mainly distinguish between the planning (a) and the execution phase (b–c) of movements. By planning, we mean all the processes required to choose the features of the movement (i.e., to *represent* the task) and by execution, the processes responsible for the spatio-temporal activation of the muscles *generating* the corresponding trajectories by the limbs. Within this framework, four different possible structures for the generation of discrete and rhythmic movements need to be considered (see Fig.3.1).

Two/Two

Discrete and rhythmic movements are generated through two totally different processes, at both the planning and the execution phase.

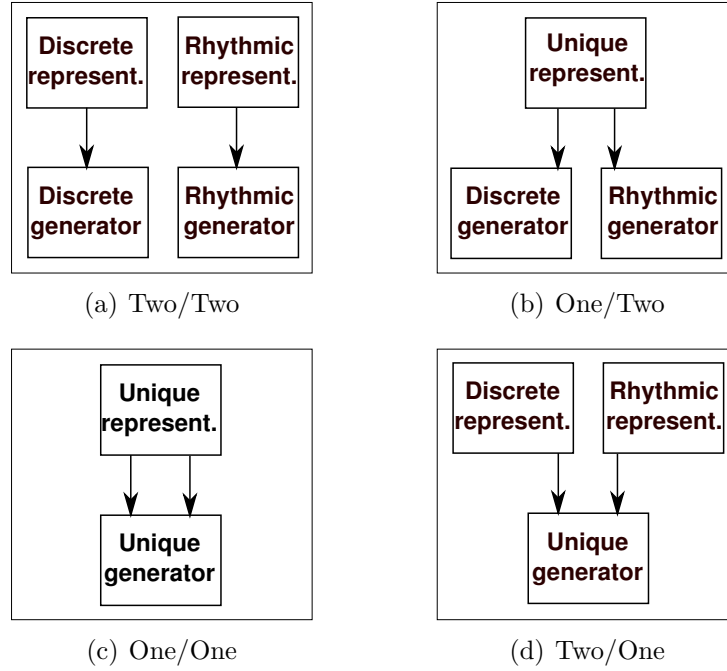


Figure 3.1: The four different categories of models.

One/Two

The planning processes involved in the generation of both movements are the same, while their generation depends on different structures.

One/One

Discrete and rhythmic movements are two outcomes of the same process, at both the planning and the execution level.

Two/One

The two movements involve different types of representations, while the generator is shared.

These four simple categories provide us with basic grounds for reflection on the possible differences between discrete and rhythmic movements. We will refer to them throughout this chapter.

3.3 Defining discrete and rhythmic movements

Mathematically, defining rhythmic and discrete movements is an easy task. Rhythmic refers to periodic signals, discrete to aperiodic signals. However, when considering movements that we actually perform, the task becomes more complex, the major problem

3.3. Defining discrete and rhythmic movements

being that movements are finite in time and that the formal, mathematical definition of periodicity is thus unusable. Moreover, the intrinsic variability of movements and modulation by the environment (contacts for instance) change the actual trajectory, so that it is impossible to perform a perfectly periodic trajectory.

The attempt by Hogan and Sternad (2007) to develop a taxonomy to classify discrete and rhythmic movements confirms the inherent difficulty of the task. A discrete movement is defined as a movement that occurs between two postures, where postures stand for a non-zero interval of time where (almost) no movement occurs. Rhythmic movements are categorized in four subsets, going from strictly periodic movements to movements with recurrent patterns. However, as the authors point out in the article, these two definitions are not exclusive. The so-called rhythmic movements occur in between postures (and thus enter the definition of discrete), and discrete movements can be repeated in order to become periodic.

Another difficulty derives from the fact that rhythmic and discrete movements have mainly been studied separately in the literature, although some interesting (relatively recent) articles on their combinations exist (as for instance Hogan and Sternad (2007) or Sternad (2007)). From our point of view, this distinction is mainly due to two interlinked factors. First, rhythmic and discrete movements have not been studied per se in general, but mainly as outcomes of some specific processes in trajectory generation, such as for instance CPGs in locomotion and sensorimotor transformations in reaching. Second, studies focusing on the low-level generation of movements often concentrate on rhythmic movements such as locomotion, while those concerning high-level movement generation typically address discrete movements such as reaching or grasping. This implies different investigation techniques; most of the studies on rhythmic movements have focused on the spinal cord–brain stem system in deafferented or spinalized subjects, whereas discrete movement is usually studied using brain imaging techniques or kinematic data on awake, behaving animals. Overcoming these differences is a necessary step to understanding discrete and rhythmic movements.

These two issues make a review of rhythmic and discrete movements difficult in the sense that any comparison between the numerous studies on the subject is laborious since the methods, the point of view and the physiological level of investigation are different. It is an interesting question whether, in terms of motor control, the apparent differences between discrete and rhythmic movements are artifacts due to different scientific approaches or if both types of movements are in fact produced independently.

Schaal et al (2004) and van Mourik and Beek (2004) for instance have defined three hypotheses that need to be considered: (a) rhythmic movements are repeated discrete movements (*concatenation hypothesis*), (b) discrete movements correspond to interrupted cyclic movements (*half-cycle hypothesis*) and (c) discrete and rhythmic movements result from different processes (*two-primitives hypothesis*). Note that these three hypotheses would correspond to the One/One case defined above for (a) and (b) and to the Two/Two case for (c). The mixed cases One/Two and Two/One are not considered here as the

planning and the execution phase of the movements are not distinguished.

While hypotheses (b) and (c) are still untested, several studies have shown that hypothesis (a) is unlikely to be true. According to van Mourik and Beek (2004), the concatenation hypothesis is mainly a consequence of trajectory planning theory where it is often supposed that discrete segments are used as building blocks for a movement. This has been ruled out by several studies comparing discrete and rhythmic movements (van Mourik and Beek (2004); Hogan and Sternad (2007)), where key kinematic features of rhythmic movements are significantly different from those of discrete movements. Schaal et al (2004) obtained similar results using fMRI techniques: some cortical areas activated during discrete movements were not active during rhythmic ones. In addition, as reported by van Mourik and Beek (2004), Guiard (1993) argued that the concatenation assumption would involve a waste of elastic energy (indeed at the end of a reaching movement, the energy has to be dissipated, whereas for rhythmic movement, the energy can be stored as potential energy for the remaining half-cycle).

It is important, however, to point out that those comparisons are always made between a reaching movement and its corresponding back- and forth- rhythmic movements. Thus the difference observed may be due to the characteristics of reaching itself (for instance the control commands required to characterize it) rather than to the fact that reaching is a discrete movement. For instance, in the experiment conducted by Schaal et al (2004), the subjects had to either cycle around a rest position at a self-chosen amplitude or to stop at a chosen position, to wait for a while and then to start again. fMRI recordings of this experiment have shown that some cortical areas active during the discrete movements were not activated during the rhythmic movements, leading to the conclusion that rhythmic movements cannot be concatenated discrete movements. However, as has been pointed out, notably by Miall and Ivry (2004), discrete movements require more processing, namely choosing where to stop and when to start again, which could also explain the difference observed in the fMRI recordings.

Another non negligible phenomenon is the onset and ending of a rhythmic movement: indeed, boundary conditions change the kinematic properties of the initial and final cycles (compared to normal, in-between cycles), making them closer to those of discrete movements. Indeed, when a discrete movement is performed, the initial and final accelerations are zero, while this is not the case during in-between cycles.

van Mourik and Beek (2004) have studied the in-between cycles and first and last half-cycles separately. They came to the conclusion that, whereas the in-between cycles were significantly different from the discrete movements, the first and last half-cycles were kinematically close to discrete movements. Even if their results do not rule out the half-cycle hypothesis conclusively, they give more support to the two-primitives hypothesis: the cyclical movements performed could in fact be a sequence in a discrete, onsetting movement, followed by rhythmic movements and terminated again by a discrete movement. A model by Schöner and Santos (2001) based on this latter hypothesis will be presented in the last part of this review.

3.4. The combination of discrete and rhythmic movements

The questions on the nature of discrete and rhythmic movements thus remain open, even if strong evidence seems to rule out the concatenation hypothesis. In the next section, we present some work on the interaction of discrete and rhythmic movements in tasks involving their combination.

3.4 The combination of discrete and rhythmic movements

Most of the EMG and kinematic studies on the combination of rhythmic and discrete movements are built on the same scheme: a particular joint (usually the finger or the elbow) has to be moved from an initial to a target position (discrete movement) while oscillating (rhythmic movement). The oscillation is either physiological (Goodman and Kelso (1983); Adamovich et al (1994); Michaels and Bongers (1994); Sternad et al (2000)) or pathological (Wierzbicka et al (1993); Elble et al (1994); Staude et al (2002)). The reader is referred to Sternad (2007) for a thorough review.

In all these experiments, an entrainment effect is observed, that is, the discrete movement is phase-coupled with the rhythmic movement, in the sense that the onset of the discrete movement occurs preferably (though not always) during a specific phase window of the oscillations. Goodman and Kelso (1983) showed that this phase window corresponds to the peak of momentum of the oscillations in the direction of the discrete movement. Interestingly, it has been shown that professional pistol shooters press the trigger in phase with their involuntary tremor, while beginners try to immobilize themselves before shooting (Tang et al (2008)).

In terms of EMG recordings, the burst initiating the discrete movement occurs approximately at the time where the EMG activity for the rhythmic movement would have been expected without this perturbation. This effect is thus referred to by De Rugy and Sternad (2003) as "burst synchronization". Performing the same experiment, although at different frequencies (lower for De Rugy and Sternad (2003)), Adamovich et al (1994) and De Rugy and Sternad (2003) came to different conclusion on movement combination. Adamovich et al (1994) observed the three following features: (a) the oscillations rapidly attenuate and disappear during discrete movements and resume after the peak velocity of discrete movements; (b) there is a phase resetting of the oscillations after the completion of discrete movements; and (c) the frequency tends to be higher after discrete movements. In addition, they observed that (d) once a discrete movement is initiated, it is performed independently of the rhythmic one, in the sense that the discrete trajectory is not influenced by the rhythmic movement. Based on the monotonic hypothesis (St-Onge et al (1993)), according to which the command of the discrete movement stops at the time of its peak velocity, they concluded that discrete and rhythmic movements are excluding each other at the neural level, in the sense that they

cannot co-occur. However, their kinematic outcomes outlast them and leads to overlap.

However, Sternad et al (2000) came to a different conclusion concerning the interdependence of the two movements. Indeed, they observed a significant influence of rhythmic movements on discrete movements (lower frequencies of oscillations lead to longer discrete movements), which is in contradiction with the result (d) obtained by Adamovich et al (1994). Moreover, the higher frequency observed by Adamovich et al. after a discrete movement (observation (c)) appeared to be a transient phenomenon. Following these observations, Sternad et al (2000) proposed that both movements co-occur and that the attenuation of the oscillations during discrete movements is due to inhibitory phenomena.

Note that co-occurrence of discrete and rhythmic movements is supported by a study on whisker movements in rats by Haiss and Schwarz (2005), where it was found that rhythmic and non rhythmic movements can be evoked through two different areas of the primary motor cortex. It was shown in addition that simultaneous activation of both areas resulted in a shift of the offset of the whisker oscillations, that is, in a combination of both movements. This experiment will be discussed in more detail in Section 3.6.

We now discuss more precisely the generation of discrete and rhythmic movements, at both the execution and the planning levels.

3.5 Generation of discrete and rhythmic movements

We present movement generation through two fundamental concepts, *CPGs* and *force fields*, that we develop in what follows.

CPGs, that is a spinal network involved in many behaviors in vertebrates and invertebrates, are a seminal concept in the generation of (rhythmic) movements (Grillner (1985), Delcomyn (1980)). Although most work on CPGs was originally dedicated to rhythmic movements, Grillner (2006) for instance now extends it to discrete movements as well.

Another important discovery in movement generation is the concept of *force fields*, which has been brought to light by Bizzi's group (Bizzi et al (1991)). As we will see, force fields provide evidence for a modular organization of the spinal cord circuitry in vertebrates.

In what follows we present these two notions in more detail, as well as their relationship to discrete and rhythmic movements.

3.5.1 Central pattern generators

Approximatively one century ago, there were two competing explanations for the rhythmic pattern present in locomotion: one suggested that sensory feedback was the main trigger of the different phases of locomotion (Sherrington (1910)), and the one suggested the existence of central neural networks capable of generating rhythms without any sensory input (Brown (1912)): such neural networks are now called CPGs.

3.5. Generation of discrete and rhythmic movements

Brown (1912) showed that cats with transected spinal cord and with cut dorsal roots showed rhythmic patterns of muscle activation. Even if the transection of the dorsal roots in the initial experiment did not exclude the role of sensory afferents, as pointed out by Grillner and Zangger (1984), there is now very clear evidence that rhythms can be generated centrally without sensory information. Indeed, experiments on lampreys (Cohen and Wallen (1980), Grillner (1985)), on salamanders (Delvolvé et al (1999)) and on frog embryos (Soffe and Roberts (1982)) have shown that when the spinal cord is isolated from the body, electrical or chemical stimulations activate patterns of activity, called fictive locomotion, very similar to those observed during intact locomotion. Since then, the CPG hypothesis has been strengthened by experiments on both vertebrates and invertebrates (see Stein et al (1997) or Ijspeert (2008) for more comprehensive reviews).

Grillner (1985) proposed that CPGs are organized as coupled unit-burst elements with at least one unit per articulation (i.e., per degree of freedom) in the body. Cheng et al (1998) reported on experiments where these units could be divided even further with independent oscillatory centers for flexor and extensor muscles. Furthermore, several experiments have shown that CPGs are distributed networks made of multiple coupled oscillatory centers (Ijspeert (2008)).

According to Marder and Bucher (2001), two types of CPG networks can be distinguished: the so-called pacemaker-driven networks and networks with emergent rhythms. Pacemaker-driven networks, which are generally always active, as in breathing, consist of a subnetwork of intrinsically oscillating neurons that drives non-bursting neurons into a cyclic pattern, while in networks with emergent rhythms, the oscillatory pattern comes from couplings between the neurons, for instance by mutual inhibition of two reciprocal neurons. A mathematical model by Matsuoka (1985) of such a system will be presented in Sect. 3.7.

While sensory feedback is not needed for generating the rhythms, it has been shown that some important features of the actual motor pattern are not present in the fictive motor pattern (Stein and Smith (2001)). For instance, in the cat scratching movement, the rhythmic alternation between agonist and antagonist muscles is already present in the fictive motor pattern, whereas the relative duration of extensor activity observed during actual scratching is greater than that observed in the immobilized preparation (fictive pattern). The motor pattern generated by CPGs thus seems to be modulated by the sensorimotor information so that it stays coordinated with body movements.

According to Pearson (2000), sensory feedback is also involved in the mechanisms underlying short-term and long-term adaptation of CPGs. He postulates that the long-term phenomena are driven by the body and limb proprioceptors together with central commands and the action of neuromodulators. Kawato (1996) also proposed that persistent errors detected by proprioceptors are used to recalibrate the magnitude of the feed forward command.

In summary, strong evidence exists for CPGs in animals, as rhythmic patterns of activation were observed both in decerebrated and in deafferented animals, the observed

pattern being thus reasonably imputed to the spinal cord alone. In humans, the activity of the isolated spinal cord is not observable, making the generalization of the previous results difficult: influences from higher cortical areas and from sensory pathways can hardly be excluded (Capaday (2002)).

Evidence suggesting that the spinal cord with intact sensory afferents can generate rhythmic locomotor-like patterns is provided by different studies on patients with complete spinal lesion (Dimitrijevic et al (1998)). In addition, Hanna and Frank (1995) reported stepping-like movements in patients before or after brain death and stepping responses have been observed in anencephalic infants just after birth (Peiper and Nagler (1963)). It was shown that treadmill exercises for patients with spinal cord injuries improved their walking pattern (Barbeau and Rossignol (1994); Dietz and Harkema (2004); Edgerton et al (2004); Rossignol et al (2007); Wolpaw and Tennissen (2001)) which may be accounted for by the fact that CPGs can be trained to function independently of descending signals (Stein (2008)). Interestingly, Dietz et al (2002) showed that in a setting with 100% body unloading (thus limiting the role of stretch reflexes), patterned leg movements could be elicited in patients with para- and tetraplegia. Moreover, studies of disabled patients have shown that in the absence of sensory information, gross movement control is preserved, even if peripheral information is necessary for precise movement organization and control (Jeannerod (1988), Gandevia and Burke (1992)).

The neonatal stepping movements are an illustration of a complex intra and inter limb coordination of muscle activity, and, even though it lacks some of the unique features of human locomotion, some of its characteristics remain with the onset of real walking, suggesting that the innate pattern could be transformed during ontogeny by neural circuits that develop later to obtain mature locomotion (Forssberg (1985))¹. Indeed, although the innate stepping response usually (but not always) disappear, the pattern used by toddlers is similar in many aspects to the patterns in newborns (Forssberg (1985); Thelen and Cooke (1987)). While Forssberg (1985) suggested that the inactive period may be due to a change of excitability in the CPG due to the developing descending locomotor driving signals, Thelen and Cooke (1987) argued that the innate CPGs evolved in a more task-specific pattern, notably through the maturation and experience of key subsystems such as balance, posture control and strength.

As mentioned above, most of the early work on CPGs focused on rhythmic movements, but the discovery of functional muscle synergies in the frog linked to discrete movements has led to an extension of the term, as we will see in the next section.

¹It should be however pointed out that the role of transient neonatal reflexes are still unclear, and in particular whether these reflexes are later used to develop mature, voluntary movements or if they correspond to different control levels.

3.5. Generation of discrete and rhythmic movements

3.5.2 Motor primitives and forces fields

The Bizzi group provided some evidence for the concept of motor primitives. Indeed, they brought to light that movements were generated in a modular way by the spinal cord in frogs (for a comprehensive review, see Bizzi et al (2008)). More precisely, stimulating specific interneuronal areas of the spinal cord, they observed that the limb was moved in the direction of the same target posture (equilibrium point) whatever the initial position of the limb was. They called the set of the vectors corresponding to the directions obtained by the stimulation *force fields*. Surprisingly, only three to four directions, corresponding to different areas in the spinal cord, were identified (Bizzi et al (1991)); furthermore, they were sufficient to account for natural limb trajectories (Kargo and Giszter (2000)).

Indeed Mussa-Ivaldi et al (1994) found that stimulating two areas simultaneously was almost equivalent to a simple linear combination of the vector of the force fields proportional to the intensity of stimulation. 87.8% (36 of 41) of the cases could be explained by the summation hypothesis, while an alternative hypothesis, where the outcome corresponded to only one of the field (i.e., a winner-take-all approach), was also tested and could explain 58.5% (24 of 41) of the cases. Under the hypothesis that the fields can be summed, and since the intensity of stimulation does not change the pattern of force orientation (Giszter et al (1993)), the space of possible end-effector target positions could be spanned through the weighted summation of a limited set of force fields. Note that similar results were obtained with rats (Tresch et al (1999)) and cats (Krouchev et al (2006); Ting and Macpherson (2005)).

The costimulation assumption supports the hypothesis that movements are produced through the combination of spinal motor primitives, which can be characterized by a resulting force field acting on the end effector of the limb. This seminal result could provide a powerful tool for explaining how the CNS can easily control the many muscles involved in any movement. Indeed, instead of having to activate and control the different muscles involved in the task, the CNS only has to define the level of activation of a small number of synergies. Furthermore, the combination being almost linear, it provides an efficient way of bypassing the inherent nonlinearities present in movement control using direct muscle activation. Tresch et al (1999) have developed a variety of computational methods to extract muscles synergies involved in different movements. Identifying those synergies is a difficult task, mainly because muscles can belong to more than one synergy at a time.

In an experiment using chemical stimulation² (NMDA iontophoresis) of interneurons in the spinal cord of the frog, Saltiel et al (1998) found that some regions were eliciting rhythmic behaviors. Force measurements of the limb show a finite number of synergies

²Although both electrical and chemical microstimulations give the same overall picture for discrete movements (see Saltiel et al (1998)), differences in the typical responses are observed that are due to the fact that electrical microstimulation excites mainly somas and axons, while chemical microstimulation excites dendrites and somas.

corresponding to the orientation of the oscillations. More precisely, in rhythmic activation, it seems that the equilibrium point changes periodically, leading to an oscillatory behavior. It is thus believed that by stimulating a particular area of the spinal cord, a whole CPG network can be activated thanks to connectivity. Interestingly, the different orientations of oscillation are very close to the direction of the force fields for discrete movements found with the same method. Furthermore, the areas of activation of the discrete and the rhythmic movements for a given orientation were topographically close (Saltiel et al (2005)). This result suggests that rhythms might arise from the temporal combination of simpler discrete modules. According to Saltiel et al (1998), CPGs could be organized such that the discrete modules provide the orientation of the oscillations while the timing features come from the network.

It is not known yet if the concept of force fields can be extended to higher vertebrates, but it has been shown that a finite set of (time-variant) synergies of muscles could account for the movement generation in humans during fast reaching movements (d'Avella et al (2006)) as well as in primate grasping (Overduin et al (2008)), providing evidence for the existence of motor primitives.

The difference between discrete and rhythmic movements, at least at the spinal level, may thus be due to differences in the topology³ of the network of motor primitives (CPGs, in the broad sense as in Grillner (2006)) rather than to completely distinct pathways. Indeed, discrete networks need to encode a target position and possibly a time of onset, while rhythmic networks also need to be endowed with a notion of frequency and phase. As reviewed by Marder and Bucher (2001), such features seem to emerge naturally from the intrinsic and synaptic properties of the neurons constituting these particular (rhythmic) CPGs.

In summary, there is strong evidence that basic building blocks of movements are present at the spinal level and that they are used by the CNS to create behaviors by combination. However, at this point it is still not clear if distinct motor primitives exist for the generation of discrete and rhythmic movement (One/Two, Two/Two cases) or if discrete and rhythmic movements are generated by the same process (One/One, Two/One cases). It seems reasonable to postulate that the same motor primitives could be involved in the generation of both discrete and rhythmic movements (by specifying target equilibrium points or orientations of oscillations, respectively), while features pertaining to rhythmic movements alone (such as frequency and phase) might arise from the coupling properties of the network.

³By network topology we mean the interconnections between the different elements of the network, including their direction and types (that is if the connection is excitatory or inhibitory in our case). Indeed, the main point is to consider the behavior emerging from the interactions between the elements (for instance a tonic or an oscillatory output), rather than the behavior of each element.

3.6 Planning of discrete and rhythmic movements

We now address the question of discrete and rhythmic movement during planning. We start by presenting the possible role of motor primitives in movement planning and then discuss movement encoding by the motor cortex.

3.6.1 Motor primitives in movement planning

A common hypothesis on how we choose to perform a given action is that the CNS uses internal models, that is, *representations* of the sensorimotor system and the environment to select the next action that it is going to produce. An inverse dynamic model is then required for movement initiation, that is, to find the activation commands to be sent to the muscles to fulfill the desired task.

The question of how the CNS actually computes the inverse model remains open. Indeed, inverse dynamics problems are complex, in particular in systems with many degrees of freedom, that is, with high redundancy. Additionally, the dynamics of the body change with time, as do external dynamics. According to some authors, the existence of motor primitives might help the CNS to solve the inverse dynamics problem (Bizzi et al (1991); Mussa-Ivaldi (1999); Georgopoulos (1996)). Indeed, motor primitives could provide the CNS with built-in links between muscles and movement direction and hence facilitate the resolution of the inverse problem of finding the muscle commands generating the desired trajectory (Mussa-Ivaldi and Bizzi (2000)).

More precisely, we have seen in Section 3.5 that motor primitives, at least in frogs, can be combined linearly, bypassing the high nonlinearity of muscles. Thus it can be imagined that instead of solving an inverse problem to control each of the muscles needed to follow the desired trajectory, the CNS chooses a combination of motor primitives that best fits this trajectory. In this case the only task of the CNS is to optimize the activation of each motor primitive in order to minimize the error between the desired and the actual trajectories. According to what was postulated in Section 3.5, such a hypothesis could mean that discrete movements are represented during planning by the CNS by a (possibly time-varying) equilibrium point in space, whereas rhythmic movements would be represented by a (possibly time-varying) direction and a parameter controlling the emerging frequency of oscillation of the network. In both cases the specification of the speed of the movement (or another, related command signal) would also be required to fully determine the movement.

Note that the existence of and the need for internal models is still debated. Basically, the opponents of internal models doubt that the brain is capable of imitating the laws of nature, which seems to be required to solve the inverse problem of finding the motor command that gives the desired kinematic outcome (for instance, the torque needed to accelerate the end effector of a limb). The reader is referred to articles by Bridgeman (2007) and Feldman (2009) for more details.

We now present some results on movement encoding that are relevant for the control of discrete and rhythmic movements.

3.6.2 Movement encoding by the motor cortex

The motor cortex can be subdivided into two areas, the primary motor cortex and the premotor cortex. The latter is formed by the lateral (dorsal and ventral) premotor areas and by the supplementary motor area, which are involved in learning sequences of movement, in timing, in the processing of sensorimotor information as well as in the selection of actions.

The primary motor cortex is involved in the control of movement parameters. According to a study by Graziano et al (2002), if the motor cortex is indeed organized somatotopically, it seems that one of the key features that is encoded in the primary cortex is the location in space towards which the movement is directed. Indeed, in their experiments, regions of the primary motor and premotor cortex of monkeys were stimulated for 500 ms (the time scale of normal reaching and grasping movements), this duration being longer than in traditional studies. They found that these stimulations were resulting in a complex movement ending in the same location, for any initial position of the limb. They concluded that, instead of encoding regions of the body, the motor cortex contains a representation of different complex postures. Note however that these results are still disputed, as reported in Strick (2002); some authors argue that the length of the stimulation and the high currents used do not ensure that only the motor cortex is activated, and thus the resulting movement may be mediated by areas other than the cortex itself.

Such a finding supports the hypothesis according to which some primary motor cortex neurons are connected in a one-to-one relationship with spinal motor synergies (Ashe (2005)); Georgopoulos (1996) has proposed a model for movement control where levels of activation of motor cortical neurons control the weights of different motor primitives at the spinal level, that is that cortical neurons elicit combinations of preprogrammed basic trajectories rather than encode the complexity of a particular desired trajectory. This could mean that the invariants observed in movement execution are the result of the usage by the CNS of a small set of motor primitives defined at the spinal level rather than a kinematic plan or optimization processes in the supra-spinal structures.

In particular, Haiss and Schwarz (2005) have studied the electric stimulation of different types of whisker movements in the rat, namely rhythmic movement (used for tactile exploration) and whisker retraction (used to sense an object at a specific location). They found that both movements, although performed by the same set of muscles, were elicited by different (but adjacent) regions of the primary motor cortex. Such a result suggests different representations for discrete and rhythmic movements (Two/One and Two/Two cases), even though it is difficult to conclude at this point whether this is due to the nature of movement (rhythmic or discrete) or simply to the fact that the motor cortex

3.7. Mathematical models

encodes behaviors (as postulated by Graziano et al (2002)). The extension of such an experiment to a broader range of movements and animals could possibly provide further insights on the differences between discrete and rhythmic movement generation.

In the same experiment, Haiss and Schwarz (2005) found that stimulating both “discrete” and “rhythmic” areas of the primary motor cortex resulted in a simple combination of the two behaviors: the resulting movement was the oscillation expected when only the rhythmic area is activated, but with an offset corresponding to the discrete movement resulting from the activation of the discrete area. This result is important as it shows that, even if discrete and rhythmic motor primitives result from different processes, which has not yet been established, the combination of those primitives still results in a coherent, meaningful behavior. A model, developed by De Rugy and Sternad (2003), representing complex movements as oscillations around time-varying offset will be presented in the next section.

3.7 Mathematical models

In this section, we illustrate the four categories (i.e., Two/Two, One/Two, One/One, Two/One) that were defined in Section 3.2 with six mathematical systems for the generation of discrete and rhythmic movements⁴.

All the mathematical models that we present here are based on dynamical system theory, that is, on sets of differential equations that define the evolution of a complex system in time. As we will see, this is a powerful approach to studying the qualitative time course of a system as well as the interconnections between its parts⁵.

Furthermore, dynamical systems are particularly well-suited for modeling discrete and rhythmic movements, as among the existing types of *stable* solutions of a dynamical system – that is, solutions robust against perturbations – two of them correspond to discrete and rhythmic signals: point attractors and limit cycles. Hence a natural solution to modeling discrete and rhythmic motor primitives is to use these stable solutions. Several examples of such modeling are presented in the following.

As a side note, combinations of stable modules are not necessarily stable themselves. However, Slotine and Lohmiller (2001) have shown that a certain form of stability, called *contraction*⁶, ensures that any combination of such contracting systems is also contracting.

⁴The code that we have used for the figures of this section is available here biorob2.epfl.ch/sd_movies, Code Chapter 3

⁵For an excellent introduction to dynamical systems, see Strogatz (2001).

⁶Contracting systems are defined as nonlinear dynamical systems in which “*initial conditions or temporary disturbances are forgotten exponentially fast*” (Slotine and Lohmiller (2001), p.138).

3.7.1 Two/Two hypothesis

In the Two/Two hypothesis, it is assumed that two different, independent processes are involved in the generation of discrete and rhythmic movements. This hypothesis is convenient for modeling because each process can be optimized in order to finely reproduce the characteristics of both discrete and rhythmic movements. Yet the question of the combination and of the mutual influence of movements is left open.

We start by presenting two independent models for discrete and rhythmic generation, developed by Bullock and Grossberg (1988) and by Matsuoka (1985) respectively. These seminal models, or extensions of them, have been used extensively in the literature [for instance in Schaal et al (2000), De Rugy and Sternad (2003) as will be presented afterwards].

- **The VITE model: a neural command circuit for generating arm and articulator trajectories**

D. Bullock and S. Grossberg,
in *Dynamic Patterns in Complex Systems*, 1988.

The VITE (*Vector Integration To Endpoint*) model was originally developed by Bullock and Grossberg (1988) to simulate planned and passive arm movements. The limb position is controlled through a neural command that modifies the respective lengths of a pair of agonist and antagonist muscles according to the desired target position.

The model thus represents a motor primitive that, given a volitional target position, controls in an automatic way a synergy of muscles so that the limb moves to the desired end state. More precisely, here the brain does not encode a trajectory, but a desired state; the actual trajectory emerges from the dynamics of the motor primitive.

The target of the trajectory of each muscle is encoded through a *difference vector*, i.e., a population of neurons representing the difference between the desired length of the muscle (T) and its actual length (p). The movement is produced by modifying the length of the muscle at a rate v (called the *activity*) that depends on the difference vector. The whole process is gated by a *go command* (G) that is a function that can modulate the speed of the movement. There are thus two control parameters, the target length T and the go command G , the output of the system being the muscle length p . Note that the function G can be chosen to be equal to a constant, a step function or a more complex signal. We will show the impact of the choice of the go command in Fig. 3.3.

Mathematical model. The following set of differential equations generates, for each muscle, a trajectory converging to the target position T , at a speed determined by the difference

3.7. Mathematical models

vector $T - p$ and the go command G :

$$\begin{cases} \dot{v} = \alpha(T - p - v) \\ \dot{p} = G \max(0, v) \end{cases}$$

where α is a constant controlling the rate of convergence of the auxiliary variable v .

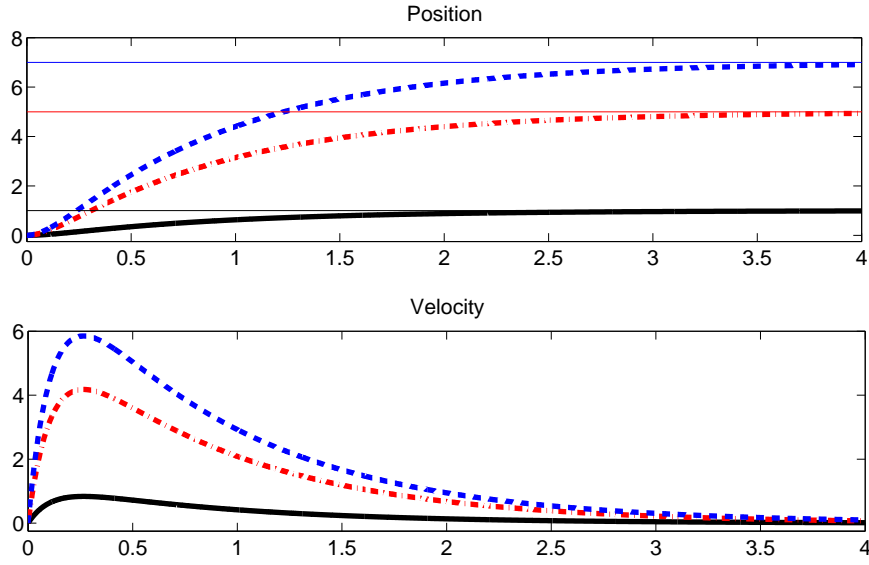


Figure 3.2: VITE model. Trajectory for three different targets: $T=1$ (plain black line); $T=5$ (dash-dotted red line); $T=7$ (dotted blue line). It can be seen that the three trajectories converge to their targets (horizontal lines) at the same time (top graph) and that the velocity peak is proportional to the displacement, i.e., to the difference vector (bottom graph). Here, for all systems, $G = 1$ and $\alpha = 10$.

As can be seen in the equations, the activity v of the population depends proportionally on the difference vector (the bigger the distance, the higher the activity and thus the speed of contraction of the muscle). In other words, the duration of the movement does not depend on the amount of contraction needed to reach the target length, but is constant, as shown in Fig. 3.2. Such a feature is very interesting when doing synchronized movements: indeed all the muscles automatically converge to their target length at the same time, whatever the difference between the target and the actual muscle length was. Moreover, this system is consistent with the observation that human pointing movements tend to have the same duration, independently of the distance that the hand has to cover [see, e.g., Morasso (1981)].

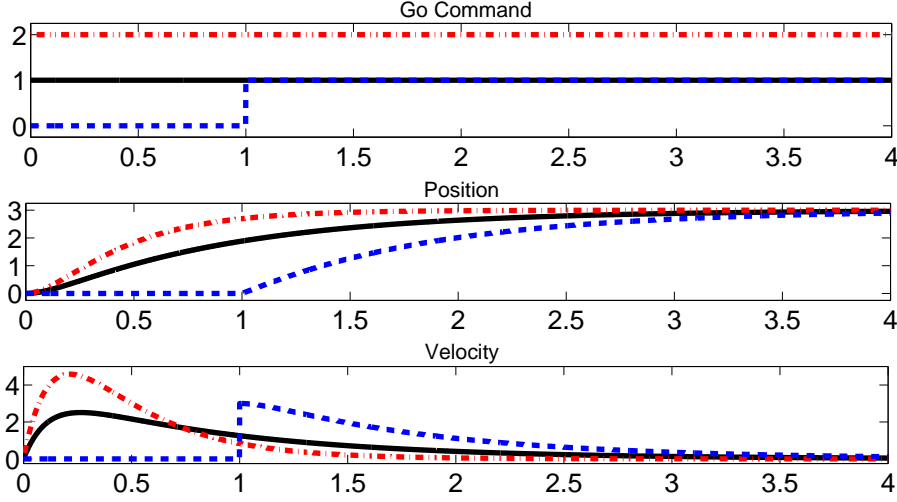


Figure 3.3: VITE model. Trajectory with three different go commands G : $G=1$ (plain black line); $G=2$ (dash-dotted red line); $G=1$ from $t=1$ s and 0 before (dotted blue line, top graph). For the three systems, the target is constant ($T = 3$). In the middle graph, it is shown that the go command can be used to postpone the onset of the movement and that the duration of the speed of convergence to the target can also be modulated. In the bottom graph, it can be seen that increasing the amplitude of the go command also increases the peak velocity. Here $\alpha = 10$.

The go command G controls both the onset of the movement and its speed profile. Indeed, once the target length T is known, nothing prevents the movement from starting except the go command (if it is set to zero). It thus allows movements to be primed before being actually executed. In addition, the amplitude of the go command G allows for a modulation of the speed defined by the difference vector. Thus the CNS can control not only the target of the movement, but also its speed. These features are illustrated in Fig. 3.3 with go commands modeled by simple step functions. Note that more complex functions can be chosen as go commands, in order to modify (and in particular smoothen) the velocity profile.

In summary, the VITE model is a very simple model for generating discrete movements with open target position and speed that allows for synchronized and delayed control of several degrees of freedom. It has been extended many times to different applications, as, for instance, for visually guided reaching movements (AVITE model, see Gaudio and Grossberg (1992)) or for modeling the interaction with the spino-muscular system to generate the torque needed to follow a specific trajectory (VITE-FLETE model, see Bullock and Grossberg (1989)).

3.7. Mathematical models

- **Sustained oscillations generated by mutually inhibiting neurons with adaptation**

K. Matsuoka,
in *Biol. Cybern.*, 1985.

In this article, Matsuoka (1985) proposes a model for oscillating neural networks. As discussed in section 3.5, it has been observed that oscillatory behaviors can emerge from networks of mutually inhibiting neurons [see, for instance, Marder and Bucher (2001)].

In Matsuoka's model, the activity of each neuron is modeled by a simple continuous-variable neuron model originally developed by Morishita and Yajima (1972). An input S_i ⁷ to the system increases the membrane potential x_i . When the membrane potential is higher than the threshold value θ , the neuron starts to fire (with firing rate y_i).

Mathematical model. The equations for one neuron are:

$$\begin{cases} \dot{x}_i = \tau(S_i - x_i) \\ \dot{y}_i = \max(0, x_i - \theta) \end{cases}$$

where τ is a parameter controlling the rate of convergence of x_i and θ is the membrane threshold.

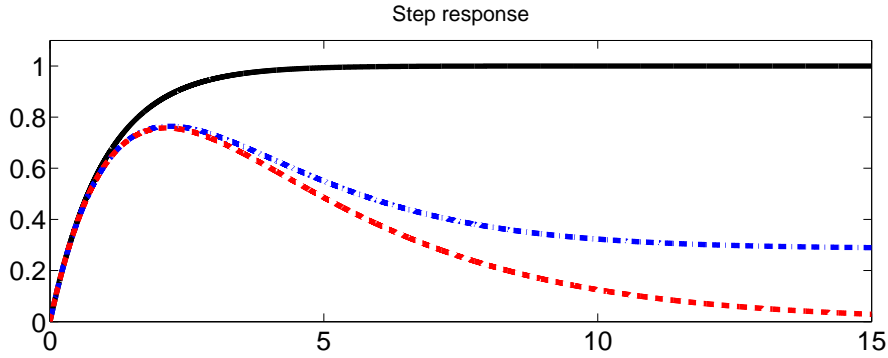


Figure 3.4: Matsuoka Oscillator. Three typical step responses of a single neuron (i.e. $S_i = 1$ in each case). *Plain black line:* fatigue parameter b is set to zero (no adaptation) and the output converges monotonically to the input value. *Dash-dotted blue line:* $b = 2.5$, the output rises but decreases after a while, showing an adaptation effect. *Dotted red line:* $b = 10^9$, and it can be seen that the firing rate almost returns to zero (which is the case when $b \rightarrow \infty$). In all cases, we used $\tau = 1$, $\theta = 0$, and $\tau' = 12b/2.5$ [this value was selected to prevent damped oscillation, see Matsuoka (1985)]

⁷Note that while we take a single value S_i as the input to the system, it can be the weighted sum of different inputs.

In this model, the firing rate increases monotonically and converges to a stationary state, which is not observed in neurons. Matsuoka (1985) thus extends the model to take into account the adaptation x' (also called fatigue) of the neurons: when the neuron receives a step input, the firing rate increases rapidly at first and then gradually decreases, as shown in Fig. 3.4. Adaptation has indeed been shown to be essential for the generation of oscillations by Reiss (1962) and Suzuki et al (1971).

Mathematical model. The model becomes

$$\begin{cases} \dot{x}_i = \tau(S_i - x_i - bx'_i) \\ \dot{x}'_i = \tau'(y_i - x'_i) \\ \dot{y}_i = \max(0, x_i - \theta) \end{cases}$$

where $\tau'(> 0)$ and $b(\geq 0)$ control the time course of the adaptation.

The neurons are then coupled to form a network. Here self-inhibition and excitation are not considered.

Mathematical model. For one neuron j , the equations are

$$\begin{cases} \dot{x}_i = S_i - x_i - bx'_i - \sum_{j \neq i} a_{ij}y_j \\ \dot{x}'_i = \tau'(y_i - x'_i) \\ \dot{y}_i = \max(0, x_i) \end{cases}$$

where the a_{ij} 's (≥ 0) are the coupling strengths of the inhibitory connections between neurons i and j and y_j is the output of neuron j . Note that here, without loss of generality, we assume $\theta = 0$ and $\tau = 1$.

Matsuoka (1985) has derived sufficient conditions for an oscillatory behavior to emerge for different types of networks. The output firing rates for two mutually inhibiting neurons are shown in Fig. 3.5.

Figure 3.6 shows two possible oscillating networks of three neurons: one where all the neurons mutually inhibit each other and the other where the neurons unilaterally inhibit each other, that is neuron 1 is, for instance, only inhibited by neuron 2 and inhibits only neuron 3.

The model offered by Matsuoka is thus a powerful tool to model different oscillatory behaviors. Note that the model can be extended to a muscle command instead of a firing rate as output; we will see an example in the model of De Rugy and Sternad (2003).

Interestingly in this model an oscillatory pattern emerges from the combination of non-cyclic units, thus reproducing the emergent rhythms observed in the spinal cord (see Section 3.5 for more details).

3.7. Mathematical models

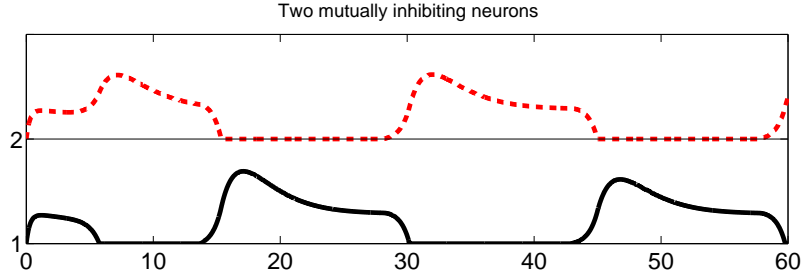


Figure 3.5: Mastuoka oscillator. The firing rate for two neurons that inhibit each other, with a constant input $S_i = 1$. Parameters were set to $a_{12} = a_{21} = 2.5$, $\tau = 1$, $\theta = 0$, $b = 2.5$, and $\tau' = 12b/2.5$

3.7.2 One/Two hypothesis

In the One/Two hypothesis, a similar encoding is used for both discrete and rhythmic movements, that is, there exists a common basic representation for the two types of movements. Such a hypothesis could reflect the analogy observed by Haiss and Schwarz (2005) between the representation of discrete and rhythmic movements in whisker movements in rats (see Section 3.6). In this model, mutual influences of movements are supposed to occur at the muscle level rather than at the spinal level, as discussed above for the Two/Two hypothesis.

We present here the model by Schaal et al (2000), in which both discrete and rhythmic movements are encoded relatively to a difference vector: between the current and desired positions for the discrete movement and between the current and desired amplitudes for the rhythmic movement.

- **Nonlinear dynamical systems as movement primitives.**

S. Schaal, S. Kotosaka and D. Sternad,

in the proc. of the *IEEE International Conference on Humanoid Robotics*, 2000

Schaal et al (2000) have developed a model based on the concept of programmable pattern generators (PPGs), that is generators of trajectories with some predefined characteristics and with some open, task-specific control parameters. Both discrete and rhythmic movements are triggered in a similar way, but they are then generated through different processes. At the end the discrete and the rhythmic output are linearly added to obtain the final trajectory.

In this model, discrete and rhythmic movements are encoded by the difference between the desired state (resp. position T and amplitude A) and the actual state (resp. p and θ); the output of the system is the position of the limb ($\alpha = p + \theta$). This system is quite complex, having many variables and parameters, so that the final output trajectory can be finely tuned to reproduce a desired movement.

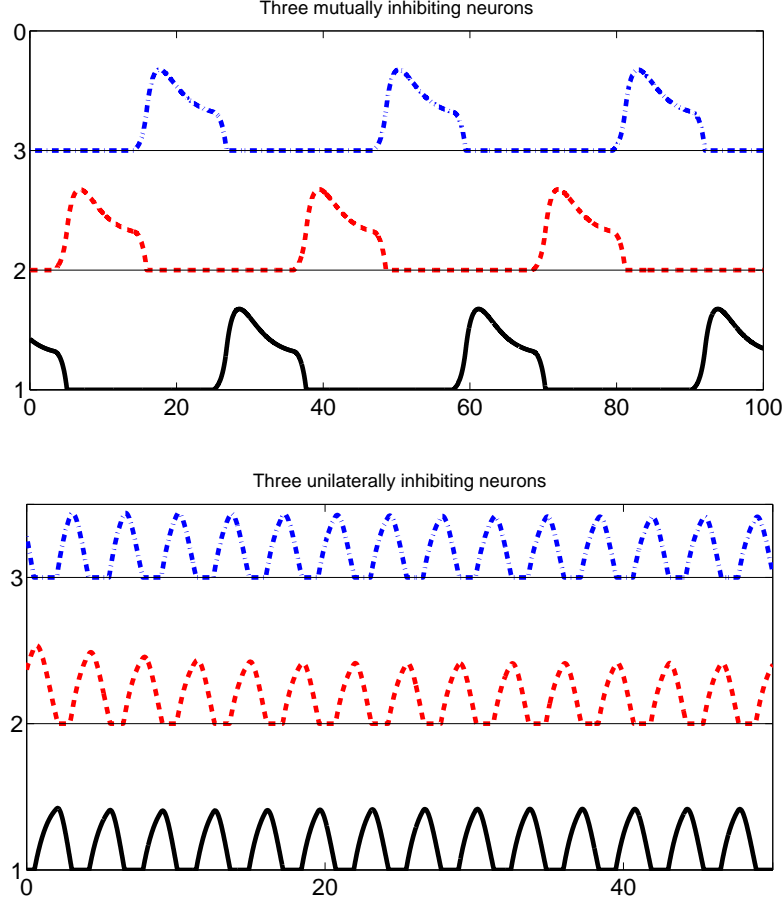


Figure 3.6: Matsuoka oscillator. The firing rate for two networks of three neurons for a constant input $S_i = 1$. *Upper graph:* the neurons are inhibiting each other, i.e., $a_{ij} = 2.5 \forall i, j = 1, 2, 3$. In the second case, the neurons are only unilaterally inhibited, i.e., $a_{12} = a_{23} = a_{30} = 2.5$ and $a_{13} = a_{20} = a_{31} = 0.0$. Other parameters were set to $a_{21} = 2.5$, $\tau = 1$, $\theta = 0$, $b = 2.5$, and $\tau' = 12b/2.5$

The discrete system is a modified version of the VITE model presented earlier. The movement of the limb is controlled through the speed of contraction of a pair of agonist/antagonist muscles. The difference vector represents the positive difference Δw_i between the desired target position of the limb T ($-T$ for the antagonist muscle) and its actual position p . Δw is then transformed into an activation pattern v_i that resembles what is observed in the primate cortex (see Fig.3.7, top panel).

Mathematical model. The difference vector for muscle i , Δw_i , is transformed into an

3.7. Mathematical models

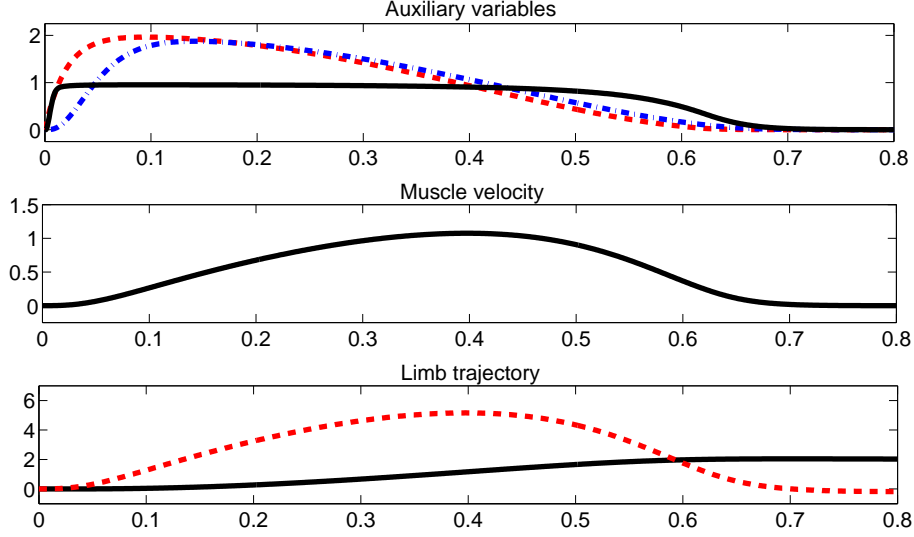


Figure 3.7: Model by Schaal et al. A typical discrete trajectory converging to the target $T = 1$. *Top panel:* activation pattern (dashed red line) as well as its smoothed version (dash-dotted blue line). The auxiliary variable r_i , which ensures that the velocity profile is roughly a symmetric, bell-shaped curve, is denoted by the plain black line. *Middle panel:* resulting speed z_i for DOF. *Bottom panel:* resulting limb trajectory (plain black line) and its speed (dashed red line). Here $a_v = 50.0$, $a_x = 1$, $a_y = 1$, $a_r = 50$, $a_z = 0.01$, $a_p = 0.08$, $b = 10$ and $c_o = 60$.

activation signal v_i

$$\begin{cases} \Delta w_i = \max(0, T - p) \\ \dot{v}_i = a_v(-v_i + \Delta w_i) \end{cases}$$

where a_v is a parameter controlling the rate of convergence of v_i .

The activation signal is then transformed into a velocity signal y_i through a double smoothing. The speed of the movement can be adjusted through the parameter c_0 .

Mathematical model.

$$\begin{cases} \dot{x}_i = -a_x x_i + (v_i - x_i)c_o \\ \dot{y}_i = -a_y y_i + (x_i - y_i)c_o \end{cases}$$

where a_y and a_x control the rate of convergence of the system, x_i is an auxiliary variable and c_0 controls the speed of the movement.

Finally, the velocity y_i is integrated in order to obtain the final desired velocity z_i for the muscle change (see Fig. 3.7, middle panel). An auxiliary variable r_i is used to make

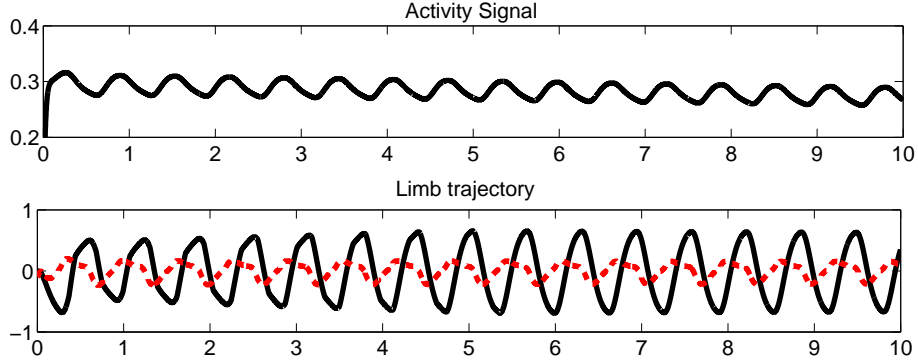


Figure 3.8: Model by Schaal et al. A typical rhythmic trajectory of amplitude $A=0.6$. *Top panel:* activation pattern ξ_i . *Bottom panel:* resulting limb trajectory (plain black line) and its speed (dashed red line). Here $a_\xi = 50.0$, $a_\psi = 1.0$, $\beta = 2.5$, $w = 2.5$, and $c_r = 20$.

z_i roughly symmetric and bell-shaped.

Mathematical model.

$$\begin{cases} \dot{r}_i = a_r(-r_i + (1 - r_i)bv_i) \\ \dot{z}_i = -a_z z_i + (y_i - z_i)(1 - r_i)c_o \end{cases}$$

where a_r and b control the shape of the signal and are chosen in order to obtain a bell-shaped velocity profile. a_z controls the rate of convergence of z_i .

The velocity commands of the agonist and antagonist muscles (i and j) are finally integrated to obtain the limb movement p (see Fig. 3.7, bottom panel).

Mathematical model.

$$\dot{p} = a_p(\max(0, z_i) - \max(0, z_j))c_o$$

where a_p controls the rate of convergence of the system and c_o its speed.

As for the rhythmic movement, it is triggered in a similar way by a difference vector $\Delta\omega_i$ between the actual amplitude θ and the desired amplitude A . $\Delta\omega_i$ is turned into an activity signal ξ_i (see Fig. 3.8, top panel).

Mathematical model.

$$\begin{cases} \Delta\omega_i = \max(0, A - \theta) \\ \dot{\xi}_i = a_\xi(-\xi_i + \Delta\omega_i) \end{cases}$$

3.7. Mathematical models

where a_ξ is a parameter controlling the rate of convergence of ξ_i .

Then, a couple of mutually inhibiting Matsuoka oscillators are used to generate oscillatory velocity signals ψ_i and ψ_j . The oscillator is slightly modified to take into account the fact that ψ_i represents a velocity and not a position.

Mathematical model.

$$\begin{cases} \dot{\psi}_i = -a_\psi \psi_i + (\xi_i + \psi_i + \beta \zeta_i + w \max(0, \psi_j)) c_r \\ \dot{\zeta}_i = -\frac{a_\psi}{5} \zeta_i + (\max(0, \psi_i) - \zeta_i) \frac{c_r}{5} \end{cases}$$

where a_ψ controls the convergence rate of the oscillators and c_r the frequency of the oscillations; w controls the strength of the inhibitory coupling.

Finally, the difference between the two oscillators (i, j) is integrated to obtain the desired trajectory θ (see Fig. 3.7, bottom panel).

Mathematical model.

$$\begin{cases} \dot{\theta}_i = \psi_i \\ \dot{\theta}_r = c_r (\max(0, \theta_i) - \max(0, \theta_j)) \end{cases}$$

where c_r controls the frequency of the oscillations.

The movement of each degree of freedom is then defined by the linear combination of the output of both signals ($\alpha = p + \theta$). This linearity allows for a simple, independent control of both movements, but it fails to reproduce the mutual influence of the discrete and rhythmic movements observed in humans.

Note that the primitives can also be coupled together in order to synchronize several degrees of freedom during coordinated movement (see Schaal et al (2000) for more details).

The many variables of the model allow for the tuning of desired basic building blocks of movements, yet also makes the system quite complex. They manage to reproduce movements containing many features reminiscent of the human generation of movement, such as a bell-shaped velocity profile, for instance.

3.7.3 One/One hypothesis

The One/One hypothesis, which assumes that a unique motor representation and generator are used to produce movements, implies either that one of the movements is a particular case of the other one (i.e., it corresponds, more or less, to the concatenation and half cycle hypotheses mentioned before) or that discrete and rhythmic movement are

themselves particular cases of a larger class of movements. The difficulty here is that the model should be designed to reproduce the mutual influences observed during movements that are both discrete and rhythmic.

We present a model developed by Schöner and Santos (2001) where discrete movements are a particular case of rhythmic ones, i.e., discrete movements are considered as truncated rhythmic movements.

- **Control of movement time and sequential action through attractor dynamics: A simulation study demonstrating object interception and coordination.**

G. Schöner and C. Santos,

in the proc. of the *9th Intelligent Symposium on Intelligent Robotic Systems*, 2001.

The model developed by Schöner and Santos (2001) was built to generate discrete movements, but it is based on limit cycles, which makes it easy to extend to the generation of rhythmic movements. Here the input is the target position T of the limb and the output is its trajectory .

In this model, discrete and rhythmic movements are both modeled using limit cycles, i.e., discrete movements are interrupted rhythmic movements. More precisely, here the attractor is a whole trajectory going from the initial position to the target position (contrarily, for instance, to the VITE model where the trajectory is a transient phenomenon and only the target position is a stable attractor). This model can thus successfully explain the observation by Bizzi et al (1984) and Won and Hogan (1995) that when a limb is perturbed during movement execution, it has a tendency to resume the original trajectory, that is, it seems that not only the target position matters, but also the trajectory leading to it.

A two-layer system is used consisting of a layer capable of generating both oscillations and stationary states ("timing layer") and another layer controlling the switching between those states ("neural dynamics control"). The timing layer consists of three terms: the first one is an attractor toward the initial state x_i , the second is a Hopf oscillator of amplitude 1 and the third is an attractor towards the target position X_f . All these terms are multiplied by the activity level of three "neurons" that are never fully active simultaneously.

Mathematical model. The equations of the timing layer are given by

$$\begin{cases} \dot{x} = -a|u_i|(x - x_i) + |u_h|(b(1 - r^2)x - \omega y) - a|u_f|(x - X_f) \\ \dot{y} = -a|u_i|y + |u_h|(b(1 - r^2)y - \omega x) - a|u_f|y \end{cases}$$

where x is the output of the system and y an auxiliary variable, and a and b control the speed of convergence of the system. In this system, $|u_j|$ ($j=i, h, f$) represents neurons that are never

3.7. Mathematical models

active (i.e., $u_j = 1$) simultaneously.

The sequence of movements is controlled by the neural layer, and more precisely through three neuron activities u_i , u_h and u_f activating the first attractor, the Hopf oscillator, and the target attractor, respectively. At rest position only the first attractor is active ($u_i = 1, u_h = 0, u_f = 0$), so that even if perturbations occur the limb stays at the same position. Then, when a command is received, the Hopf oscillator is activated ($u_h = 1$) and the first attractor deactivated ($u_i = 0$), so that the trajectory follows the limit cycle until it is close enough to the final target. At this moment the Hopf neuron activity u_h is set to zero and the final attractor is activated ($u_f = 1$) so that the trajectory converges to the target position X_f . This sequence of actions is illustrated in Fig. 3.9.

Mathematical model. The timing of activation of the three "neurons" is controlled by the neuronal dynamics which are given by the following equations:

$$\begin{cases} \alpha \dot{u}_i = \mu_i u_i - |\mu_i| u_i^3 - c(u_h^2 + u_f^2) u_i \\ \alpha \dot{u}_h = \mu_h u_h - |\mu_h| u_h^3 - c(u_i^2 + u_f^2) u_h \\ \alpha \dot{u}_f = \mu_f u_f - |\mu_f| u_f^3 - c(u_i^2 + u_h^2) u_f \end{cases}$$

Each equation corresponds to the normal form of a degenerate pitchfork bifurcation controlled by parameters⁸ μ_i with an extra term to ensure that only one neuron is active, i.e. that any solution with more than one neuron active is destabilized. The parameters μ_i are given by

$$\begin{cases} \mu_i = 1.5 + 2b_i \\ \mu_h = 1.5 + 2(1 - b_i)(1 - b_f) \\ \mu_f = 1.5 + 2b_f \end{cases}$$

where $b_i = 1$ is equal to 1 when no movement occurs and is set to 0 to activate the movement, and

$$b_f = 1 - \tanh(10(0.7X_f - x_r(i))) + 1)/2.$$

Movements can thus be shaped through the neuronal dynamics that qualitatively change the space of solutions of the timing layer. The trajectory in three parts produced by this model (i.e., discrete, rhythmic, discrete) is analogous to the observation by van Mourik and Beek (2004) that the first and last half-cycles of a rhythmic movement resemble a discrete movement. In systems with multiple degrees of freedom, coordination can be obtained through the coupling of rhythmic parts of the system [see Schöner and Santos (2001) for more details]. Synchronized discrete movements can be obtained through coupling.

⁸That is the system has one stable solution ($u = 0$) when μ_i is negative and two stable ones ($u_i = 1$ and $u_i = -1$) when μ_i is positive.

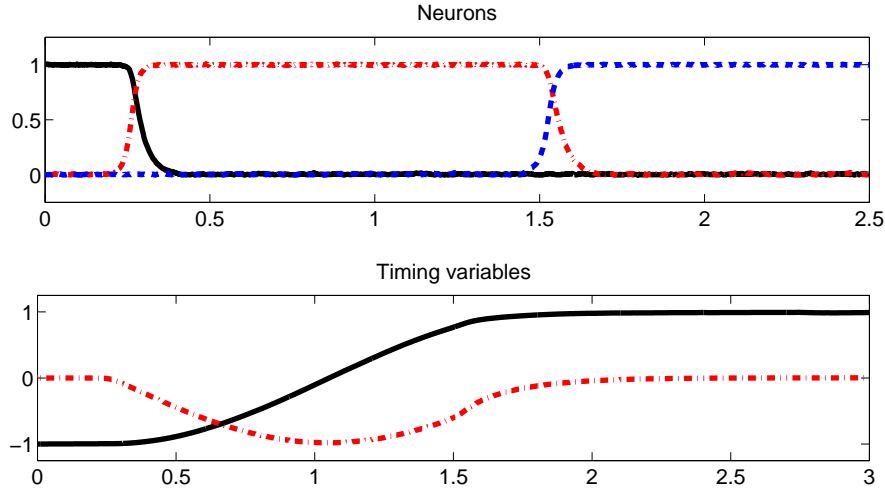


Figure 3.9: Model by Schoener and Santos. *Top panel:* activity of three neurons (u_i , plain black line; u_h , dash-dotted red line; u_f , dashed blue line) during a typical discrete movement can be observed. Only one neuron is active at a time, corresponding to three stages of the movement: rest at initial position, move to the target, and rest at the target position. *Bottom panel:* obtained trajectory x_i is shown (plain black line) as well as the auxiliary variable y_i . Here $a = 5$, $b = 1$, $\omega = 2$, $c = 2.1$, and $\alpha = 0.02$.

3.7.4 Two/One hypothesis

In the Two/One hypothesis, two different motor commands are sent to the same generator. An open question is then how the two motor commands are combined. We present here a model developed by De Rugy and Sternad (2003), initially to explain the phase entrainment effect, where both commands are simply summed.

- **Interaction between discrete and rhythmic movements: reaction time and phase of discrete movement initiation during oscillatory movements.**
A. de Rugy and D. Sternad,
in *Brain Research*, 2003

This model was originally developed to explain the phase entrainment effect observed in humans (see Sect. 3.4 for more details). Here a motor command S , composed of the sum of a discrete S_d and a rhythmic S_r command inputs, is sent to a two-neuron Matsuoka oscillator to generate two firing rates (x_i, x_j). These firing rates are then transformed into muscle commands ($\mathcal{T}_i, \mathcal{T}_j$) for a pair of agonist-antagonist muscles and finally to a limb trajectory θ .

3.7. Mathematical models

The discrete command is modeled as a pulse followed by an exponential decay, resulting in a damped oscillation that, with well-tuned parameters, will later generate a discrete movement. The rhythmic command is simply a constant signal.

Mathematical model. The input command is given by

$$S = S_r + S_d$$

where $S_r = \text{const}$ and

$$\dot{S}_d = \tau_s(-S_d + p_d)$$

where p_d is the peak value of the pulse and τ_s a time constant.

A network of two mutually inhibiting Matsuoka oscillators is then used to transform this neural command S into the firing rates (x_i, x_j) of two motoneurons controlling a pair of agonist-antagonist muscles.

Mathematical model. The network is governed by the following equations (for one neuron):

$$\begin{cases} \dot{x}_i = \tau_1(-x_i - \beta x'_i + S - \omega \max(0, x_j)) \\ \dot{x}'_i = \tau'(-x'_i + \max(0, x_i)) \end{cases}$$

where τ and τ' are two parameters controlling the time course of, respectively, the firing rate x_i and the fatigue (or self-inhibition) x'_i , β is the gain of the fatigue component and x_j is the output of the second neuron.

The firing rates of the neurons (x_i, x_j) are then transformed into torques $(\mathcal{T}_i, \mathcal{T}_j)$ exerted by a pair of agonist-antagonist muscles.

Mathematical model. The torques are obtained through the following equations:

$$\begin{cases} \mathcal{T}_i = h_{\mathcal{T}} \max(0, x_i) \\ \mathcal{T}_j = -h_{\mathcal{T}} \max(0, x_j) \end{cases}$$

where $h_{\mathcal{T}}$ is the gain of the torques.

Finally the action of the torques on the movement of the joint θ is deduced from the dynamics of the limb.

Mathematical model. The dynamics of the limb is governed by the following equation:

$$I\ddot{\theta} + \gamma\dot{\theta} - (\mathcal{T}_i + \mathcal{T}_j) = 0$$

where I is the inertia of the limb and γ is its damping.

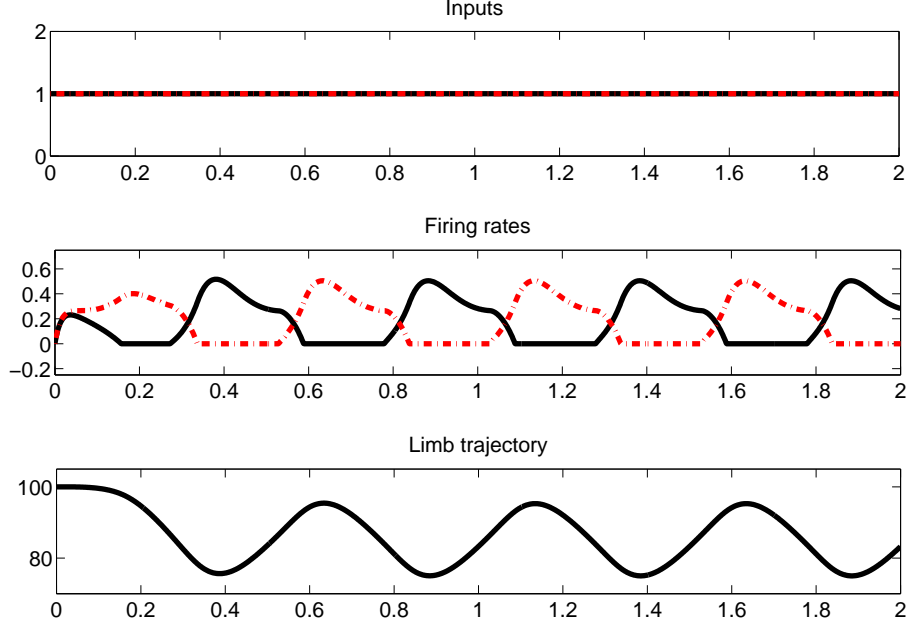


Figure 3.10: Model by De Rugy and Sternad. A purely rhythmic command $S = S_r = 1$ (top panel) leads to oscillations of the coupled neurons (middle panel) and the limb (bottom panel). Here $\gamma = 0.5$, $I = 0.08$, $h = 5$, $\tau = 0.05$, $\tau' = 0.125$, $\tau_s = 0.2$, $\beta = 2.5$, and $\omega = 2.5$.

Figure 3.10 illustrates the output of the model for a rhythmic command (that is, a constant input). The oscillating firing rates are transformed into a smooth, sinusoidal trajectory through the dynamics of the limb. In Fig.3.11, it is shown that a purely discrete movement can be obtained using a peak motor command. Finally, in Fig. 3.12, the combination of both command signals and the resulting, combined trajectories are shown.

In this model, there is an entrainment effect that emerges from synchronization effects between the two Matsuoka neurons. The distribution of the offset, as well as the phase lag observed in human subjects was successfully reproduced by this model (De Rugy and Sternad (2003)). Note that this model has been extended by Ronsse et al (2009) to integrate reafferent signals, and thus to capture bimanual features.

3.7.5 Discussion of the models

We have presented different mathematical models, whose principal characteristics are summarized in Table 3.1. All these models are based on the concept of motor primitives,

3.7. Mathematical models

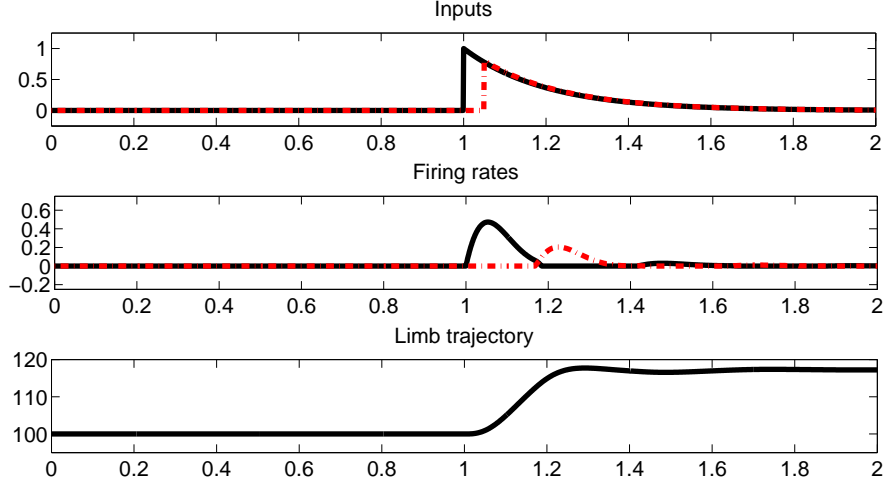


Figure 3.11: Model by De Rugy and Sternad. A purely discrete command $S = S_d$ of peak $p_d = 1$ (top panel) leads to strongly damped oscillations of the neurons (middle panel), resulting in a discrete movement of the limb (bottom panel). Here $\gamma = 0.5$, $I = 0.08$, $h = 5$, $\tau = 0.05$, $\tau' = 0.125$, $\tau_s = 0.2$, $\beta = 2.5$ and $\omega = 2.5$.

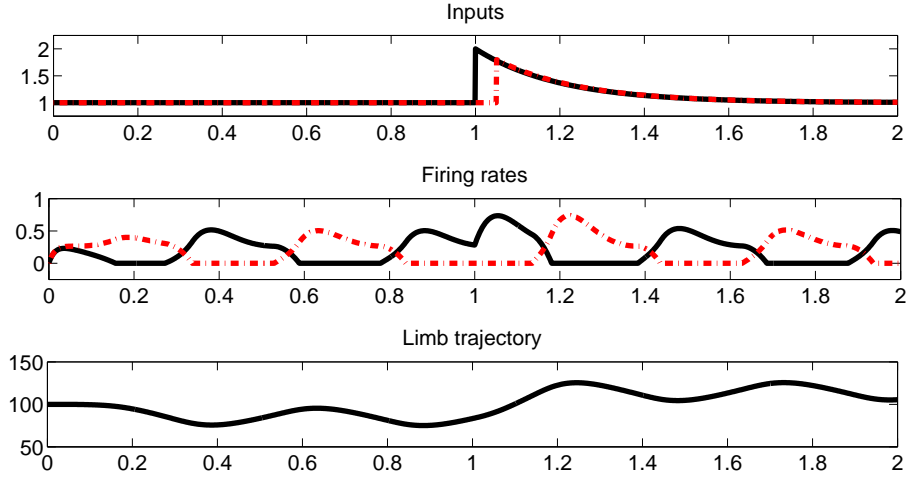


Figure 3.12: Model by De Rugy and Sternad. A combined command $S = S_r + S_d$ with $S_r = 1$ and $p_d = 1$ (top panel) leads to a perturbed oscillatory behavior of the neurons (middle panel), resulting in a rhythmic movement around a varying offset (bottom panel). Here $\gamma = 0.5$, $I = 0.08$, $h = 5$, $\tau = 0.05$, $\tau' = 0.125$, $\tau_s = 0.2$, $\beta = 2.5$, and $\omega = 2.5$.

that is, simple, non patterned commands from the brain are turned into complex output trajectories governed by the dynamics of the system. So even though the outputs of the models are not at the same representation level, they can easily be modified to account for another level of representation [for instance, De Rugy and Sternad (2003) apply the model of firing rates of neurons of Matsuoka (1985) to limb control by extending the system to the muscles and the limb dynamics].

Model	Category	Type	Ctrl	Var	Param
Bullock et al.	Two/Two	D	2	2	1
Matsuoka	Two/Two	R	n	$3n$	$3n + n(n - 1)$
Schaal et al.	One/Two	D+R	2	26	13
Schoener et al.	One/One	$D \subset R$	2	7	8
De Rugy et al.	Two/One	DR	1	5	6

Table 3.1: Main properties of the different models. *Type* refers to the type of movements and their relationship: D = discrete only, R =rhythmic only, $D+R$ = discrete and rhythmic as a linear combination of the generator outputs, DR = discrete and rhythmic as a unique generator output, $D \subset R$ = discrete as truncated rhythmic. *Ctrl* is the number of high-level commands needed to specify the movement, *Var* is the number of variables and *Param* is the number of parameters of the system. For the Matsuoka model, n refers to the number of neurons involved in the network.

All these models are successful in producing more or less complex discrete and rhythmic trajectories (except for the models of Matsuoka and Bullock, which only model one type of movement). However, in order to be plausible, these models should also be able to reproduce the interaction observed in humans between discrete and rhythmic movements mentioned in Sect. 3.4. As stated earlier, there are two main studies on the subject by Adamovich et al (1994) and Sternad et al (2000), and they come to different conclusions. While they both agree that

- (a) the rhythmic movement is inhibited by the discrete one;
- (b) the phase of the rhythmic movement is reset after the discrete one;
- (c) the frequency tends to be higher after the discrete movement (transient phenomenon according to Sternad et al (2000));

Adamovich et al (1994) conclude that

- (d1) the discrete trajectory is not influenced by the rhythmic movement.

which is refuted by Sternad et al (2000), since they observe that

- (d2) the rhythmic movement influences the discrete one, or more precisely lower frequencies of oscillations lead to longer discrete movements.

3.8. Concluding remarks

To rule out either the Two/Two–One/Two or the One/Two–One/One categories, an efficient way to proceed would be to determine whether the mutual influence between discrete and rhythmic movements appears at the spinal or at the muscular level, i.e., if the discrete and the rhythmic dynamics influence each other because there is a unique spinal motor primitive generating them or if it is an artifact due to overlaps during the actual production of the movement. More precisely:

- In both the Two/Two and One/Two hypotheses, the question of the combination of the two movements is left open; more precisely, the interaction has to happen at a lower level of the generation process, that is at the muscular level, as proposed for instance by Adamovich et al (1994) or by Staude et al (2002). Adamovich et al (1994) postulate that discrete and rhythmic movement cannot co-occur, i.e., that any movement can be seen as a sequence of discrete or rhythmic movements. According to them, the mutual influence observed is due to the overlapping of the kinematic outcome of the two movements: they postulate that the kinematic outcome of a movement lasts longer than its generation. Note that this view is not shared by Sternad et al (2000), as discussed earlier (see Sect. 3.4). Staude et al (2002), for their part, propose that complex movements arise from the summation of the two movements subject to a threshold-linear mechanism; it is interesting to note that this simple model manages to model the entrainment effect presented in Sect. 3.4 [see Staude et al (2002) for more details].
- In the One/One hypothesis, the distinction between discrete and rhythmic movements is assumed to be an artifact of movement categorizations, both movements being in fact generated through the same process. In these models, the notion of interaction of the two movements is an ill-posed problem, as they indeed are produced by the same process. In this model, the mechanisms listed above should thus emerge from the dynamics of the system.
- In the Two/One hypothesis, only the representation of the movements is different, the process generating them being the same. In this case, as in the One/One hypothesis, the observed mutual influence should emerge from the dynamics of the motor primitives, as, for instance, the entrainment effect in the model by De Rugy and Sternad (2003).

3.8 Concluding remarks

Synergies of muscles have been observed in vertebrates (as reviewed in Sect. 3.5), which indicates that movement may be built through the combination of spinal building blocks of movements that we call motor primitives. Such an assumption has strong implications

for the analysis of discrete and rhythmic movements, in the sense that the intrinsic difference between them may lie at the spinal level rather than in the high-level commands used to encode them. Indeed, evidence has been presented that both discrete and rhythmic movements could result from spinal motor primitives elicited by simple, non-patterned brain commands, suggesting that the two types of movements may simply emerge from a difference in the topologies (oscillatory or not) of the spinal network underlying them.

Since we have chosen to take a functional approach, most of the results that we have presented come from animal studies. Even if these results cannot necessarily be generalized to humans in a straightforward way, we believe that they can provide insights into the processes underlying discrete and rhythmic movement generation in humans.

In this chapter, we have shown that the concept of motor primitives is an interesting approach to the question of the generation of discrete and rhythmic movements and its modeling, notably because it leads to the definition of four categories of models for movement generation. Such categories provide a framework for the analysis of the ins and outs of different approaches to the generation of discrete and rhythmic movements and thus to discard or corroborate these approaches.

Further experimentations would be required to better understand which category is the closest to reality and which are the intrinsic advantages of this model over the others. For instance, studying the switch between purely discrete and purely rhythmic tasks in humans could help to understand issues related to continuity. Experiments, where a discrete task with is performed with one hand and, simultaneously, a rhythmic task with the other hand, could bring to light the existence of a coupling between the two types of tasks. Tasks involving more than one degree of freedom could also provide information on whether the coupling occurs at the joint or at the end-effector level. Finally, experiments similar to the ones performed by Sternad et al (2000) and Adamovich et al (1994), but on wider range of frequencies and durations for the discrete movements could help to conclude on the contradictory results reported in these two articles.

As it is impossible for us at this point to conclude about the most plausible model, and as our main focus is on robotic application, we decided to develop in this thesis a model corresponding to the One/One category. Indeed, using a unique representation

- eases the computations when movements are combined;
- simplifies the switch between behaviors (continuity),

and a unique generator

- intrinsically embeds the two dynamics; and
- eases the integration of feedback.

The next chapter will present in details the mathematical model that we developed.

CHAPTER 4

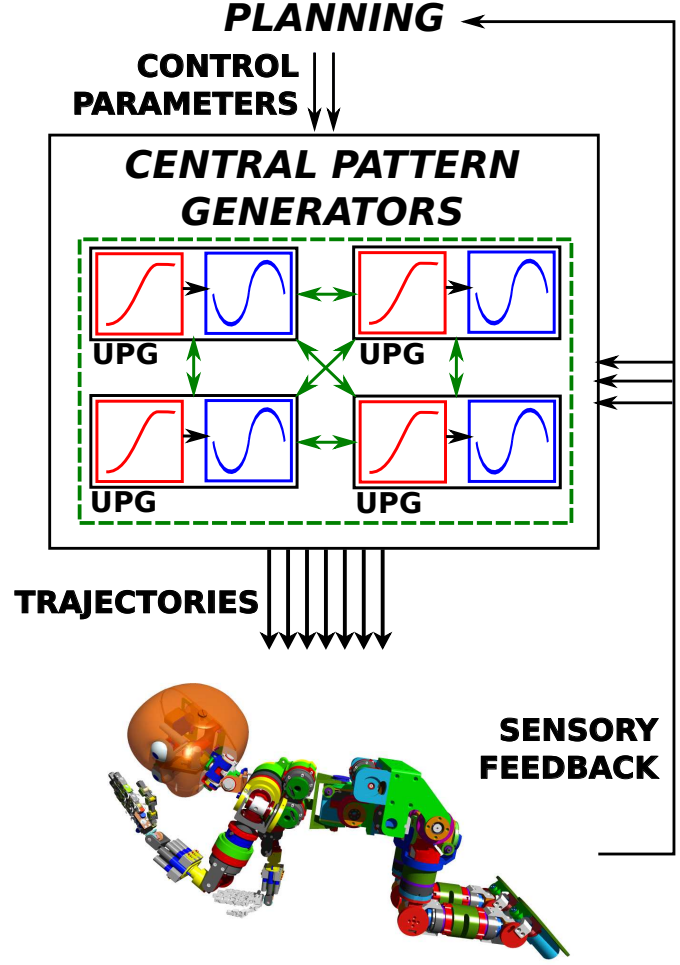
CONTROL ARCHITECTURE

Vertebrates are able to quickly adapt to new environments in a very robust, seemingly effortless way. To explain both this adaptivity and robustness, we have seen in the previous chapter that a very promising perspective in neurosciences is the modular approach to movement generation: Movements results from combinations of a finite set of stable motor primitives organized at the spinal level. In this chapter, we apply this concept of modular generation of movements to the control of robots with a high number of degrees of freedom, an issue that is challenging notably because planning complex, multidimensional trajectories in time-varying environments is a laborious and costly process. We thus propose to decrease the complexity of the planning phase through the use of a combination of discrete and rhythmic motor primitives, leading to the decoupling of the planning phase (i.e. the choice of behavior) and the actual trajectory generation. Such implementation eases the control of, and the switch between, different behaviors by reducing the dimensionality of the high-level commands. Moreover, since the motor primitives are generated by dynamical systems, the trajectories can be smoothly modulated, either by high-level commands to modulate the current behavior or by sensory feedback information to adapt to environmental constraints.

After a brief introduction of the general organization of the control architecture, we present the discrete and the rhythmic systems separately in Section 4.2 and 4.3 respectively and we discuss their combination in Section 4.4. We then present how to couple the dynamics to create a CPG (Section 4.5). We end this chapter with a brief discussion of our model compared to those presented in the previous chapter, Section 3.7.

This chapter is an extended version of the work presented in Degallier et al (2010) (submitted). Note that previous versions of the control architecture have been presented in Degallier et al (2008), Degallier et al (2007) and Degallier et al (2006). The Matlab code that we used for this chapter is available here: biorob2.epfl.ch/sd_movies, Code Chapter 4.

Figure 4.1: Schema of the control architecture. A central pattern generator is seen here as a network of dynamical systems that allows for the generation of complex output trajectories given simple, non patterned inputs. The output of the system can be further modulated by sensory information. *In the dash line box:* The discrete and the rhythmic systems are combined together to form a unit pattern generator (UPG) that is responsible for the control of one degree of freedom (DOF). The UPGs of each DOF are then coupled together in a network, the central pattern generator (CPG, in green), in order to generate a coordinated behavior between the DOFs. Note that while the (open-loop) dynamics of the UPG is always the same, the CPG depends on the structure of the robot and on the task to be accomplished.



4.1 Introduction

The general schema of the control architecture that we are proposing here is depicted on Figure 4.1: the CPGs generate trajectories according to the control parameters specified by the planning system, the whole architecture being influenced by feedback information coming from the robot. Note that this concept is not limited to control in the joint space and can easily be extended to operational space control. In our CPG, we model all movements through the combination of a discrete and a rhythmic motor primitives, both produced by a unique dynamical system (that we call a unit pattern generator (UPG)). More precisely, movements are modeled as oscillatory movements around time-varying offset. Purely discrete movements can be obtained by setting the amplitude of the oscillations to zero and purely rhythmic ones by setting a constant offset.

We now present the precise implementation of the central pattern generators. As illustrated on Figure 4.1, all trajectories (for each joint) are generated through a unique

4.2. Discrete System

set of differential equations, that we call a unit pattern generator (UPG) and which is designed to produce complex movements modeled as periodic movements around time-varying offsets. Each UPG can be divided into two subsystems: the discrete and the rhythmic ones. The first subsystem is responsible for the generation of short-term, goal directed features of the movement and the second subsystem for periodic features of the movements such as amplitude and frequency of the pattern. The dynamics of the different DOFs can then be embedded in a larger network (the CPG) through coupling to ensure coordinated and synchronized behaviors.

4.2 Discrete System

To generate discrete movements, we use a set of differential equations based on the VITE (Vector Integration To Endpoint) model originally developed by Bullock and Grossberg (1988) to simulate planned and passive arm movements. The target of the trajectory of each muscle is encoded through a *difference vector* that represents the difference between the desired position of the DOF (γ_i) and its actual position (y_i). The speed of the movement is controlled by the so-called activity v that is proportional to the difference vector ($y_i - \gamma_i$). Such an implementation allows for a coordinated control of several DOFs, as the time of convergence to the target is independent of the length of the trajectory, that is all DOFs will attain their target position simultaneously even if the distances to be covered by the joints are different (see Figure 4.2(a)). The original system by Bullock and Grossberg (1988) was slightly modified to ensure that the initial speed of a movement is zero and that the velocity profile is bell-shaped. More precisely, for each degree of freedom i , a goal directed movement towards a target position γ_i can be generated through the following set of equations:

$$\dot{h}_i = 1 - h_i \quad (4.1)$$

$$\dot{y}_i = v_i \quad (4.2)$$

$$\dot{v}_i = -\frac{1}{4}B^2h_i^2(y_i - \gamma_i) - Bh_iv_i \quad (4.3)$$

where y_i is the output of the system, v_i and h_i are auxiliary variables and B is a constant that controls the time of convergence of the system¹. The system is critically damped so that the output y_i of Eqs 4.2 and 4.3 converges asymptotically and monotonically to the target γ_i with a speed of convergence controlled by B .

The parameter h_i was added to the system to have a null acceleration at the onset of the movement. The variable h_i acts directly on the speed of convergence B , so that the system is critically damped for any value of h_i [see 4.2.1 below]. We want that $h_i = 0$

¹Throughout this dissertation, Greek letters will denote control parameters, lower-case Latin letters variables and capital Latin letters constant values

at the onset of the movement ($Bh_i=0$ and $\dot{v}_i=0$) and $h_i = 1$ ($Bh_i = B$) at the end, with a time evolution given by Eq 4.1 (the *go command*), where this particular equation has been chosen for its simplicity. h_i is reset to zero before the onset of a new movement; a movement is considered to be new when the difference between the new target and previous one is large, indicating a new movement rather than a correction of the current one (the threshold was set to 0.1 rad in our case).

This system is relatively simple in the sense that the only parameter to be tuned is the rate of convergence B , and the trajectory is fully determined by simply specifying one control parameter: the target γ_i of the movement.

4.2.1 Stability and analytical solution

To ensure the stability of the system, we can analyze the eigenvalues of its Jacobian, that is

$$J_D = \begin{pmatrix} -1 & 0 & 0 \\ 0 & 0 & 1 \\ -0.5B^2h_i(y_i - \gamma_i) & -0.25B^2h_i^2 & -Bh_i \end{pmatrix}$$

Thus $\det(J_D - \lambda I) = (-1 - \lambda)(\lambda + 0.5Bh_i)^2$, and hence $\lambda_0 = -1$ and $\lambda_1 = \lambda_2 = -0.5Bh_i$. For the go command, we have:

$$h(t) = 1 - e^{-t+t_0}.$$

where t_0 is the time of initiation of the movement, as we set $h(t_0) = 0$ in our case. Hence $0 \leq h_i \leq 1$ and $B > 0$ and thus the general system is stable as all the eigenvalues are negative. The two eigenvalues of the system given by Eqs. 4.2-4.3 are equal and real and hence the system is critically damped. Thus, if we consider $h_i = 1$, the solution is given by

$$y(t) = \gamma + C_y e^{-\frac{B}{2}t} + C_v t e^{-\frac{B}{2}t}$$

where C_y and C_v are constant that depends on the initial conditions y_0 and \dot{y}_0 .

4.2.2 Some Interesting Properties

We now present some features of the system – illustrated in Figure 4.2 – that will be useful for the application to robotics.

Globally attractive fixed point

The fixed point γ_i is globally attractive, which means that the trajectory will asymptotically converge to this point for any initial condition, as illustrated in Figure 4.2(a)). Moreover, as mentioned above, for any initial condition, all trajectories converge to the target γ_i at the same time, as the speed is proportional to the remaining distance to be covered, as can be observed on Figure 4.2(a). Such a feature is interesting because all the DOFs move in a synchronized way, the

4.2. Discrete System

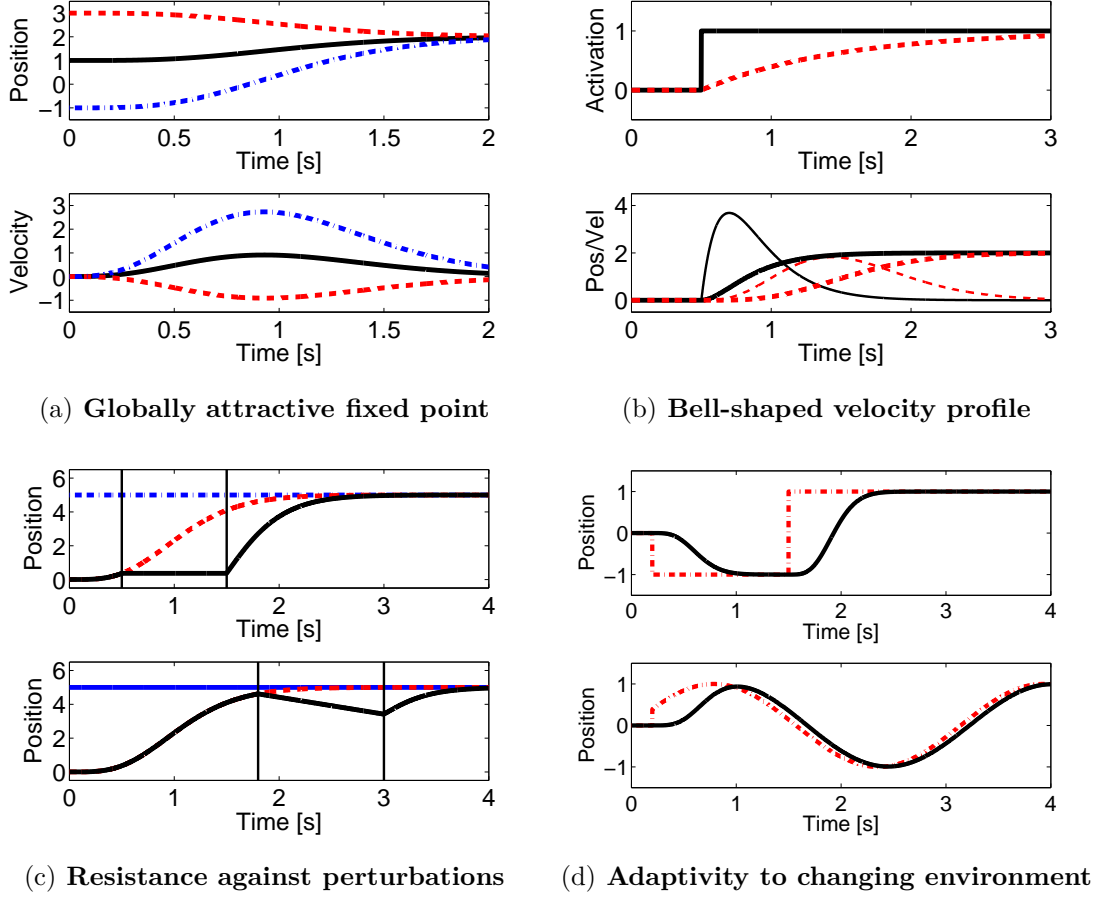


Figure 4.2: Discrete System. See text for discussion. The system was integrated using Euler method with a time step of $t = 0.001$ s. Here the gain in Eq. 4.3 is set to $B = 10$. **(a)** *Top panel:* Different trajectories converging to same target position $\gamma_i = 2$ with different initial positions: $y = 1$ (black, plain line), $y = 3$ (red, dash line) and $y = -1$ (blue, dash dotted line). *Bottom panel:* Corresponding velocity profiles (same color/line type). **(b)** *Top panel:* Two types of activation command h_i : in black, plain line a step response ($h_i = 1$ at the time of activation $t = 0.5$ s, 0 before) and in red, dotted line a monotonically increasing activation (corresponding to the output of Eq.4.1). *Bottom panel:* Resulting velocity profiles with the constant activation (black, plain thick line) and with the increasing activation (red, dotted thick line) and the corresponding trajectories (same color/line type but thin lines) converging to the target $\gamma_i = 2$. **(c)** *Top panel:* The normal trajectory (in red, dash line) is modified (in black, plain line) due to a perturbation where the DOF is kept in a constant position ($y = \text{cst}$, in-between the vertical lines) from $t = 0.5$ to $t = 1.5$, but eventually converges to the target $\gamma_i = 5$ (in blue). *Bottom panel:* In this case, the perturbation is similar to a "force" exerted on the DOF ($\frac{dy}{dt} = -1$, in-between the vertical lines) from $t = 1.8$ to $t = 3.0$ (same color code/line types as in the top panel). **(d)** *Top panel:* The target position γ_i (in red, dash dotted line) is changed from $\gamma_i = -1$ to $\gamma_i = 1$ before convergence. In black, plain line is the resulting trajectory. *Bottom panel:* The time varying target position $\gamma_i(t)$ is here given by a sine signal (same color code as in the top panel). Note that here $B = 50$ to illustrate the fast convergence to the target.

drawback being that the speed of the movement is not directly controlled (unless B is changed).

Bell-shaped velocity profile

The auxiliary variable h_i modifies the velocity profile of the system to make it bell-shaped. More specifically, it is used to decrease the acceleration at the onset of the movement. The effect of the chosen activation compared to a simple step response (as in the original VITE model) is illustrated on Figure 4.2(b). Note that the auxiliary variable h_i must be reset to zero at each onset of a new movement.

Resistance against perturbations

Thanks to the global attractiveness of the fixed point, even if a short term perturbation occurs during or after the transient – as illustrated on Figure 4.2(c) – the trajectory will eventually converge to the target position. This feature is interesting because it can be used to modify the trajectory according to sensory information: for instance, if the DOF is stuck in a given position due to a (temporary) obstacle for instance, the dynamics of the system can be temporarily modified so that the desired trajectory matches the actual environmental condition by using a perturbation similar to the one shown in Figure 4.2(c) (top panel). Similarly, a repulsive force can be applied to avoid contact with a (temporary) obstacle (Figure 4.2(c), bottom panel). Application of such feedback strategies during drumming will be presented in Section 5.

Adaptivity to changing environment

Figure 4.2(d) illustrates the ability of the system to smoothly adapt to changes of the target position γ_i . In the top panel, it is shown that if the target position is suddenly changed (if for instance the object that has to be reached is suddenly moved, or if the target object changes), the trajectory is smoothly modulated to converge to the new target position. The bottom panel of Figure 4.2(d) depicts the case where the target position γ_i is constantly changed. In this case the system cannot converge to the desired position (at least with our choice of convergence rate B), but is constantly updated so that it reproduces the trajectory of the moving target with a time delay. In order to deal with a constantly changing target position, the activity command h_i is reset only when the difference between the new target and the previous one is big enough (the threshold was set to 0.1 rad in our case). Application of this feature will be shown both in drumming (Section 5) and crawling (Section 6).

4.3. Rhythmic system

4.3 Rhythmic system

For the rhythmic system, we use a modified Hopf oscillator. Indeed, such an oscillator has many interesting properties, among which: (i) it has a unique periodic solution that is globally stable, (ii) this solution can be found analytically and is a perfect sine, and (iii) the frequency and the amplitude are explicit parameters. The system can be written as:

$$\dot{m}_i = C(\mu_i - m_i) \quad (4.4)$$

$$\dot{x}_i = \frac{A}{|\mu_i|} (m_i - r_i^2) x_i - \omega_i z_i + n \quad (4.5)$$

$$\dot{z}_i = \frac{A}{|\mu_i|} (m_i - r_i^2) z_i + \omega_i x_i + n \quad (4.6)$$

where x_i is the output of the system, y_i and m_i auxiliary variables, $r_i = \sqrt{x_i^2 + z_i^2}$, A and C are constant controlling the rate of convergence and n is a noise signal distributed normally ($n \sim \mathcal{N}(0, 1)$) added to avoid unstable solutions. The first term of the right-hand side of Eqs. 4.5 and 4.6 ensures convergence to a desired constant amplitude while the second term induces the oscillatory behavior. When $\mu_i > 0$, Eqs. 4.5 and 4.6 describe an Hopf oscillator whose solution x_i is a sine of amplitude $\sqrt{\mu_i}$ and frequency ω_i . A Hopf bifurcation occurs when $\mu_i = 0$ leading to a system with a globally attractive fixed point at (0,0) when $\mu < 0$ (see Fig. 4.3). Note that Eq.4.4 was added to the canonical system to ensure that this bifurcation is smooth.

4.3.1 Stability and analytical solution

To analyze the system, we rewrite the oscillator in polar coordinates $(r, \theta)^2$ for x and z :

$$\dot{m} = C(\mu - m) \quad (4.7)$$

$$\dot{r} = \frac{A}{|\mu|} (m - r^2) r \quad (4.8)$$

$$\dot{\theta} = \omega \quad (4.9)$$

with $r \in \mathcal{R}^+$ and $\theta \in \mathcal{R}$. In this way the radius and the phase dynamics are decoupled. The solutions of Eqs. 4.7 and 4.9 are straightforward:

$$m(t) = \mu - (\mu - M_0)e^{-Ct} \quad (4.10)$$

$$\theta(t) = \omega t + \Theta_0 \quad (4.11)$$

²We do not follow here the convention stated before (see footnote 1 of this chapter) according to which Greek letters denotes control parameters, since the Greek letter θ is commonly used to denote the variable corresponding to the phase.

where $M_0 = m(0)$ and $\Theta_0 = \theta(0)$. It can easily be seen that μ is a stable fixed point. The phase θ is increasing at a constant rate. It means that perturbations will not be forgotten, but will also not increase. Eq.4.8 bifurcates depending on the value of μ (as m will eventually converge to μ), indeed for $\mu \leq 0$ the system has a unique solution $r = 0$, while for $m > 0$, it has two solutions, $r = 0$ and $r = \mu$, as illustrated on Fig. 4.3.

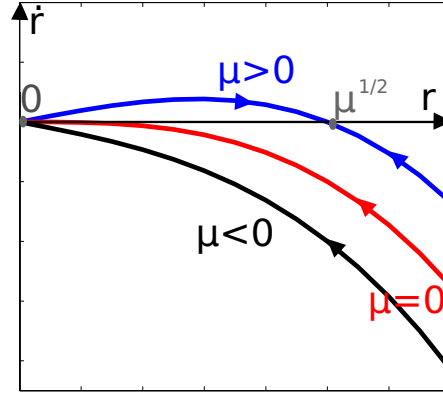


Figure 4.3: Hopf bifurcation. Depending on the value of μ , the solutions of system change qualitatively. If $\mu > 0$ (in blue in the figure), the system has two solutions $r = 0$ and $r = \sqrt{\mu}$. In this case, and as indicated by the arrows that shows the direction of trajectories, $r = 0$ is a “repeller” and $r = \sqrt{\mu}$ an attractor. However, for $\mu = 0$ (in red) and $\mu < 0$ there is only one solution left ($r = 0$) and it is attractive.

If we consider that $m(t) = \mu$, we can solve the system for the non-zero solution by using the fact that Eq.4.8 is a Bernoulli equation. We obtain:

$$r^2 = \frac{\mu}{1 + \mu C_r e^{-\frac{2A}{|\mu|}(\mu t)}} \quad (4.12)$$

where C_r is a constant depending on the initial conditions.

4.3.2 Some Interesting Properties

We now present some features of the system – illustrated in Figure 4.4 – that will be useful for the application to robotics.

Attractive limit cycle

As illustrated on Figure 4.4(a), all trajectories will eventually converge to the limit cycle for any initial conditions. Indeed, the system has two solutions, a stable limit

4.3. Rhythmic system

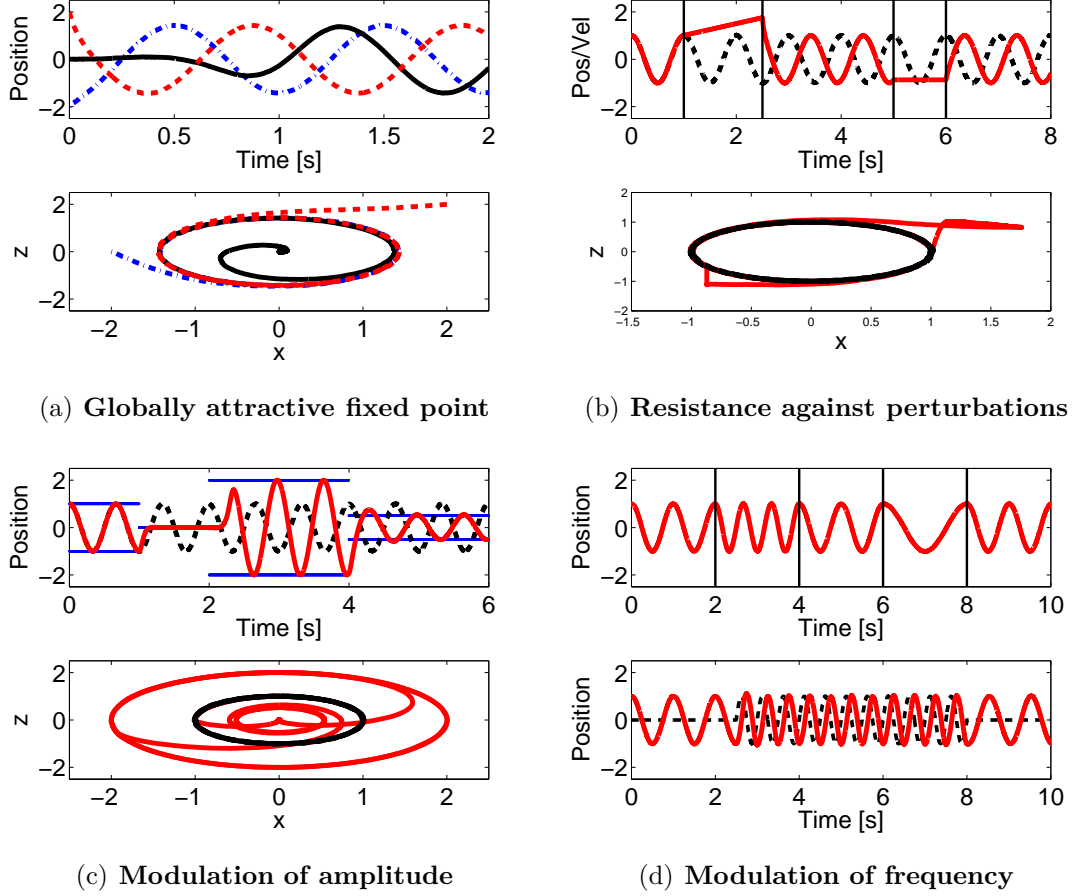


Figure 4.4: Rhythmic System. See text for discussion. The system was integrated using the Euler method with a time step of $t = 0.001$ s. The gain in Eqs. 4.5 and 4.6 is set to $A = 5$, and the gain in Eq.4.4 is set to $C = 20$. The noise ε is distributed normally with mean 0 and standard deviation 1. **(a) Top panel:** Different trajectories converging to same limit cycle of amplitude $\sqrt{2}$. with different initial positions: $x = 0, z = 0$ (black), $x = 2, z = 2$ (red) and $x = -2, z = 0$ (blue). **Bottom panel:** The same trajectories in the phase plane- xz (same color). **(b) Top panel:** The normal trajectory (black) is modified (red) due to a perturbation where the DOF is kept in a constant position ($x = \text{cst}$) from $t = 0.5$ to $t = 1.5$ and from $t = 1.8$ to $t = 3$. **Bottom panel:** The same trajectories in the phase plane- xz (same color). **(c) Top panel:** The initial trajectory (black) is modulated through the parameter m (in blue $\pm\sqrt{m}$) resulting in changes in amplitude (red). At $t = 1$, m is set to a negative value (-5), leading to a Hopf bifurcation: the limit cycle becomes a fixed point system. At $t = 2$, m is set to 4 and the reverse bifurcation occurs. **Bottom panel:** The same trajectories in the phase plane- xz (same color). **(d) Top panel:** Modulation of the parameter ω_i : at $t = 0\text{s}, \omega_i = 2\pi$, at $t = 2\text{s}, \omega_i = 3\pi$, at $t = 4\text{s}, \omega_i = 2\pi$, at $t = 6\text{s}, \omega_i = \pi$ and at $t = 8\text{s}, \omega_i = 2\pi$ (the black vertical lines denote times where ω_i is changed). **Bottom panel:** The original signal (red, plain line) is entrained by a signal $F(t) = \sin(4\pi)$ (black, dotted line) with a gain equal to 10, i.e. $\dot{x} = \dots + 10F$, from $t = 2.5\text{s}$ to $t = 8\text{s}$.

cycle (a circle centered at the origin and of radius $\sqrt{\mu_i}$) and an unstable fixed point at $(0,0)$. Thus, thanks to the noise added in the equation (n), the system will eventually converge to the oscillatory solution even if initially at the unstable fixed point. However, as illustrated in Figure 4.4(a), the convergence might be slower in that case (black, plain line) than for any other initial condition (blue, dashed and red, dotted-dashed lines). Note that the noise does not affect the general profile of the curve as it randomly affects the velocity during extremely short period during the integration process.

Resistance against perturbations

Thanks to the attraction of the limit cycle, even if a short-time perturbation occurs, the system will resume to the limit cycle afterwards, as depicted on Figure 4.4(b). Similarly to the discrete case, this feature can be used to modulate the dynamics of the system according to feedback information. For instance, it will be used to control the transition between swing and stance in crawling (see Section 6).

Modulation of amplitude and Hopf bifurcation

The amplitude of the oscillation is directly controlled by the parameter μ_i , more precisely, the amplitude is equal to $\sqrt{\mu_i}$ (when $\mu_i > 0$). This feature allows us to very easily and smoothly modulate the system behavior according to the desired trajectory output, as illustrated in Figure 4.4(c). When $\mu_i < 0$, the space of solutions of the system qualitatively changes: it consists of a unique stable fixed point at the origin (instead of a stable limit cycle and an unstable fixed point). This phenomenon is called a Hopf bifurcation. In Figure 4.4(c) Hopf bifurcations occur at $t = 1$ and $t = 2$. Thanks to the addition of Eq. 4.4, both transitions are smooth. Note that without the addition of noise, the transition from the fixed point solution to the limit cycle can be very slow, as the fixed point remains a solution (even if unstable) after the bifurcation.

Modulation of frequency

Similarly to the amplitude, the frequency can be smoothly modulated directly through the parameter ω_i , as shown in Figure 4.4(d), top panel. Note that a periodic perturbation, if strong enough, can induce entrainment, i.e. the overall frequency of the oscillator will synchronize to that of the external signal, as can be seen on Figure 4.4(d), bottom panel. We will see in Subsection 4.5 that entrainment between oscillators can be used to couple them together.

4.4. Unit pattern generator

4.4 Unit pattern generator

In order to develop a controller that can generate both discrete and rhythmic movements, we superimpose the dynamics of the two systems presented before in order to obtain a limit cycle that can be moved in the x -direction (as depicted of Figure 4.5(b), bottom panel), i.e. the discrete movement is applied as a translation of the rhythmic one. This is obtained by embedding the discrete movement output y_i as an offset of the rhythmic output x_i , that is

$$\dot{h}_i = 1 - h_i \quad (4.13)$$

$$\dot{y}_i = v_i \quad (4.14)$$

$$\dot{v}_i = -\frac{1}{4}B^2h_i^2(y_i - \gamma_i) - Bh_iv_i \quad (4.15)$$

$$\dot{m}_i = C(\mu_i - m_i) \quad (4.16)$$

$$\dot{x}_i = \frac{A}{|\mu_i|} (m_i - r_i^2) (x_i - y_i) - \omega_i z_i + n \quad (4.17)$$

$$\dot{z}_i = \frac{A}{|\mu_i|} (m_i - r_i^2) z_i + \omega_i (x_i - y_i) + n \quad (4.18)$$

where x_i is the output of the system and now $r_i = \sqrt{(x_i - y_i)^2 + z_i^2}$. When $\mu_i > 0$, Eqs. 4.5 and 4.6 describe a Hopf oscillator whose solution x_i is a periodic signal of amplitude $\sqrt{\mu_i}$ and frequency ω_i with an offset given by γ_i . A Hopf bifurcation occurs when $\mu_i < 0$ leading to a system with a globally attractive fixed point at $(\gamma_i, 0)$. The set of equations Eqs. 4.14-4.18 is a *unit pattern generator* (UPG), that is the minimal set of equations controlling one degree of freedom, while Eq. 4.13 can be shared by several DOFs to ensure synchronized discrete movements.

	γ_i	μ_i	ω_i
D	non constant	negative	any
R	constant	positive	non zero
D+R	non constant	positive	non zero

Table 4.1: Types of movements. This table summarizes the influence of the control parameters on the type of the movement. Here D = purely discrete, R = purely rhythmic, D+R = a combination of rhythmic and discrete movements.

Figure 4.5(a) (black line) depicts the qualitative behavior of the system depending on parameters μ_i and γ_i : the system can switch between purely discrete movements (from $t \approx 1s$ to $t \approx 2s$), purely rhythmic movements (from $t \approx 2s$ to $t \approx 5s$), and combinations of both (from $t \approx 6s$ to $t \approx 7s$), the control parameters being extremely simple as can be seen from the top panel. Discrete movements are simply elicited by specifying the target

position γ_i (in blue), while rhythmic movements are controlled through the specification of the parameter μ_i (in red), which is the square of the amplitude of the output movements. Table 4.1 summarizes the control parameters and the induced types of behaviors.

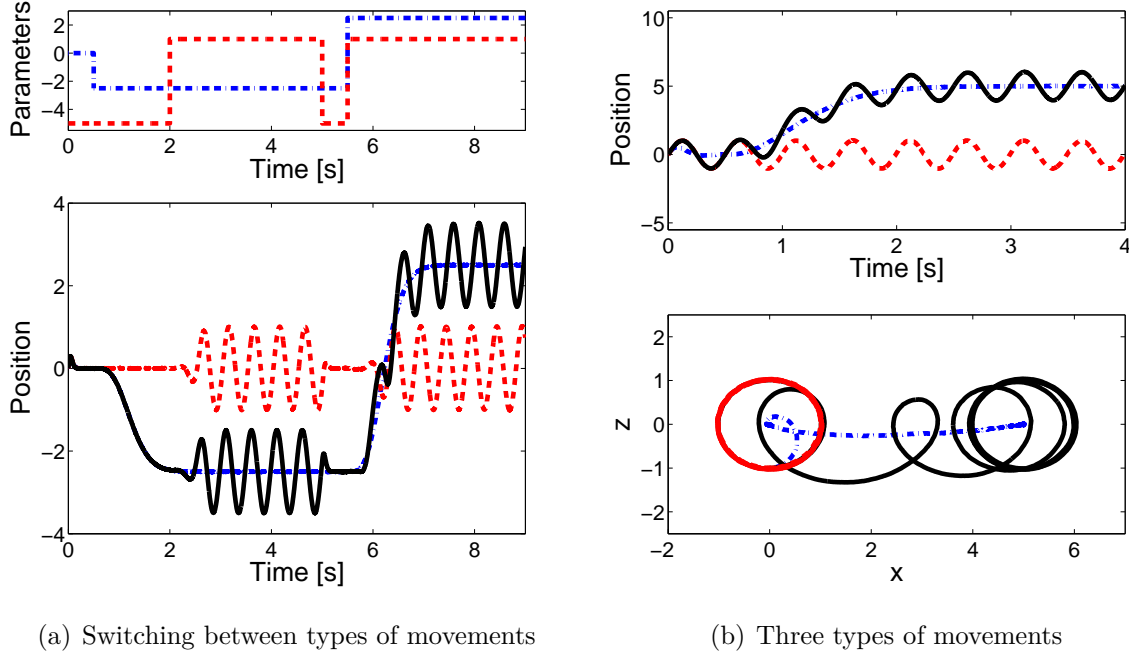


Figure 4.5: Unit pattern generator. See text for discussion. We used $B = 10$, $A = 5$, $C = 20$, $\omega_i = 4\pi$ and Euler integration (time step $t = 0.001$). **(a) Top panel:** The control parameters: in red, dash line, the amplitude, in blue, dash-dotted line, the target of the movement. **Bottom panel:** In black, plain line is the trajectory corresponding to the control commands of the top panel, in red, dash line, the movements resulting when no discrete movement is elicited ($\gamma_i = 0$) and in blue, dash-dotted line, when the rhythmic movement is switched off ($\mu_i = -5$). **(b) Top panel** A purely discrete movement in blue, dash-dotted line, a purely rhythmic one in red, dash line, and the combination of both in black, plain line. **Bottom panel:** The corresponding trajectories in the phase plan (same color/line code).

The control of each degree of freedom is thus defined by a set of 6 equations (one of which – Eq. 4.13 – can be made common to all the DOFs to ensure a synchronized onset for the discrete movements), 3 internal constant parameters (A , B and C) and 3 control parameters (γ_i , μ_i and ω_i). This implementation is thus economic, as the target γ_i (respectively the amplitude μ_i and the frequency ω_i) are the minimal information needed to characterize a discrete (rhythmic) movement. Note that the parameter B can be used as a control parameter to modulate the speed of the discrete component of the movement, but we will keep it constant in all the applications.

4.5. Central pattern generator

4.5 Central pattern generator

In order to obtain a coordinated behavior between several DOFs, their UPGs can be coupled in a network to obtain coordinated behaviors. Such networks, that we call central pattern generators (CPGs), ensure fixed time relationships between the different rhythmic outputs (i.e. phase-locking), a feature which is particularly convenient for generating different gaits for locomotion, as illustrated on Figure 4.6.

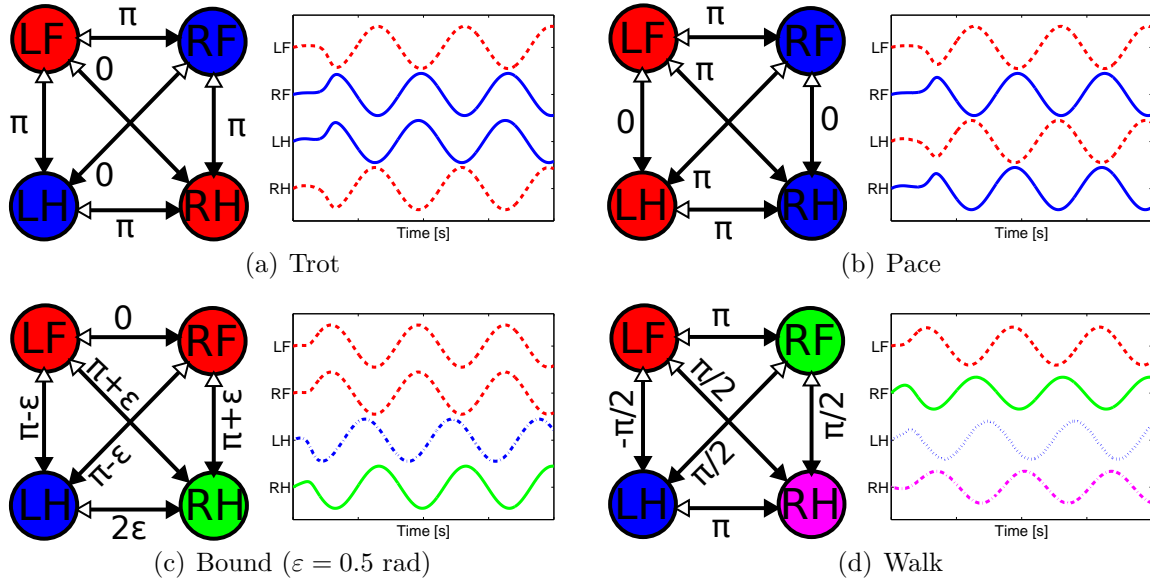


Figure 4.6: CPGs applied to Gait Generation. See text for discussion. LF = left forelimb, RF = right forelimb, LH = left hind-limb and RH = right hind-limb. Here $K_{ij}^{x/z} = 8, \forall i, j$ and we used $B = 10, A = 5, C = 20, \omega_i = 4\pi$. The system was integrated using Euler method with a time step of $t = 0.001$ s. **(a-d)** Schemes of the phase shifts for the different gaits (*left*) and the corresponding trajectories (with same color) (*right*). Cells/trajectories of the same color means that they are in phase. Note that for each arrow, the angle θ attached is the angle corresponding to the full, black arrow, whereas the angle corresponding to the white arrow should be taken as the opposite ($-\theta$) to ensure coherence. For a more explicit specification of the angles, please refer to Table 4.2.

The coupling of DOF i with other DOFs (j 's) is done by extending Eqs. 4.17 and 4.18 in the following way

$$\dot{x}_i = \dots + \sum_{j \neq i} K_{ij}^x (\cos(\theta_{ij})(x_j - y_j) - \sin(\theta_{ij})z_j - (x_i - y_i)) \quad (4.19)$$

$$\dot{z}_i = \dots + \sum_{j \neq i} K_{ij}^z (\sin(\theta_{ij})(x_j - y_j) + \cos(\theta_{ij})z_j - z_i) \quad (4.20)$$

where θ_{ij} is the desired phase difference between DOF i and j and the $K_{ij}^{x/z}$'s are the

Trot	LF	RF	LH	RH	Bound	LF	RF	LH	RH
LF	0	π	π	0	LF	0	0	$\pi - \varepsilon$	$\pi + \varepsilon$
RF	$-\pi$	0	0	π	RF	0	0	$\pi - \varepsilon$	$\pi + \varepsilon$
LH	$-\pi$	0	0	0	LH	$-\pi + \varepsilon$	$-\pi + \varepsilon$	0	-2ε
RH	0	$-\pi$	0	0	RH	$-\pi - \varepsilon$	$-\pi - \varepsilon$	2ε	0
Pace	LF	RF	LH	RH	Walk	LF	RF	LH	RH
LF	0	π	0	π	LF	0	π	$-\pi/2$	$\pi/2$
RF	$-\pi$	0	$-\pi$	0	RF	$-\pi$	0	$\pi/2$	$\pi/2$
LH	0	π	0	π	LH	$\pi/2$	$-\pi/2$	0	π
RH	$-\pi$	0	$-\pi$	0	RH	$-\pi/2$	$-\pi/2$	$-\pi$	0

Table 4.2: Angles needed for different gaits. This table summarizes the angles required to generated the different gaits presented on Figure 4.6. See text for discussion. LF = left forelimb, RF = right forelimb, LH = left hind-limb and RH = right hind-limb. ε is an open parameter that controls the phase shift between the two hind legs in the bound gait.

(constant) gains of the coupling, i.e. the rate of convergence to a stable solution. The matrix $\Theta = (\theta_{ij})$ is skew-symmetric with a null diagonal, i.e. $\theta_{ij} = -\theta_{ji}$ and $\theta_{ii} = 0$, $\forall 1 \leq i, j \leq n$, if n is the number of coupled DOFs.

The coupling is dissipative, i.e. we subtract the current state of the system. In this way, the coupling term is null when the system has converged to the desired solution. It means that the coupling influences the dynamics only when needed, i.e. during the transient states.

4.5.1 Some Interesting Properties

CPGs design

Thanks to the couplings, a network with fixed relationships between the different elements can be designed. Figure 4.6 depicts CPGs corresponding to the following gaits: trot, pace, bound and walk, and the corresponding trajectories. As shown in Table 4.2, the matrix of θ_{ij} 's is skew-symmetric and it has a null diagonal. Note that the CPG network should be designed in a coherent way, in the sense that the sum of every phase differences along a closed path must be a multiple of 2π so that a cell is required to be in phase with itself.

Smooth online modulation

The phase relationships between the different elements of the CPG can be modified online as illustrated on Figure 4.7. Note that the time required to converge to the new solution depends on the parameters k_{ij}^x and k_{ij}^z . Similarly, if a short-term perturbation occurs, the system will resume to the desired phase-shift relationship afterwards.

4.6. Concluding remarks on the control architecture

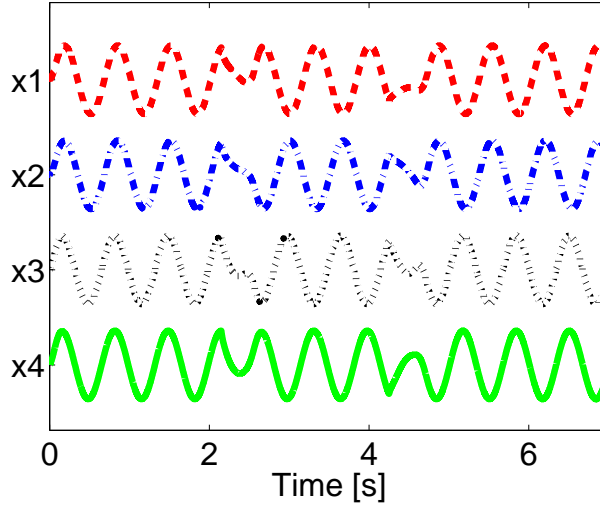
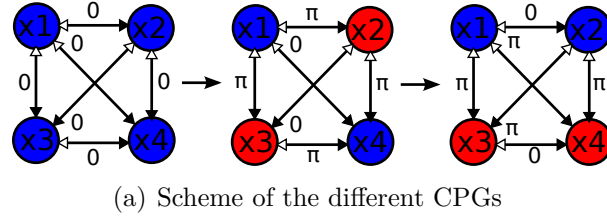


Figure 4.7: Transition between different couplings. See text for discussion. Here $k_{ij}^{x/y} = 5$, $\forall i, j$ and we used $b = 10$, $a = 5$, $\omega_i = 4\pi$. The system was integrated using Euler method with a time step of $t = 0.001$. (a) Scheme of the different CPGs configurations and (b) the corresponding trajectories.

4.6 Concluding remarks on the control architecture

We use the combination of two types of stable solutions of dynamical systems – fixed points and limit cycles – to model complex movements as oscillatory movements around time-varying offsets, a null amplitude corresponding to a purely discrete movement and a fixed offset to a purely rhythmic one. In this way, a wide range of behaviors, from purely discrete tasks such as reaching to coordinated rhythmic behavior such as locomotion, can be generated by the controller, and moreover, the controller can easily switch from one behavior to another as the same system is used for all types of movements. To ease the planning phase, the CPG was designed so that the open parameters of the dynamical system are the key characteristics of the movements, which are the target final position for discrete movements and the amplitude and frequency for rhythmic ones. As was illustrated, the main advantages of dynamical systems are that (i) the solutions obtained

can be designed to be robust against perturbations, and that (ii) trajectories can be modified smoothly in real time according to high level command or sensory feedback and finally that (iii) the system can be entrained by external signals and thus several systems can be coupled to obtain synchronized behaviors. In addition, the integration process requires low computational power and thus can be easily used for fast control loops.

These properties will be illustrated in the next two chapters through two applications: interactive drumming (Chapter 5) and crawling and reaching (Chapter 6).

CHAPTER 5

APPLICATION TO DRUMMING

Drumming is a challenging application as it requires coordination between the limbs, precise timing and the robust online modulation of the parameters of the movement. Drumming has been implemented on robots several times before, to study agent-object interaction (Williamson (1999)), learning from demonstration (Ijspeert et al (2002)) or human-robot interaction (Kose-Bagci et al (2010)) for instance. Here our goal is to show the adaptability and robustness of the architecture: trajectories are modulated online both by high level commands and by feedback information. A first implementation is done on the HOAP-2, which is improved later on in a second implementation on the iCub.

First implementation The HOAP-2 robot is playing with its two arms on two bongos and a cymbal. The robot can play any score up to the frequency limit imposed by the motors. In simulation, the target angle positions of the arms are constantly updated according to the actual positions of the drums through an inverse kinematics algorithm, while on the real robot these angles are manually defined beforehand.

Second implementation On the iCub robot, the four limbs as well as the head are controlled. The robot plays on an electronic drums, with 4 drum pads and two pedals. A graphical user interface (GUI) was developed to allow any user to define the score to be played by the robot on the fly, but predefined scores can also be used. A feedback loop based on collision detection is used to better handle the contacts between the sticks and the electronic drums by temporarily stopping the limb in its current position every time a collision occurs. In addition, a visual tracker, developed by S.Gay (Gay et al (2010)), is used to detect the position of the drums and to adapt the movement of the limbs to their actual position.

This work was done in collaboration with Ludovic Righetti (BIOROB, EPFL, Suisse, now USC, USA), Sebastien Gay (BIOROB, EPFL, Suisse), Francesco Nori (IIT, Italy), Lorenzo Natale (IIT, Italy) and Cristina Santos (UMINHO, Portugal) (see Annex C for a

detailed list of the contributions). More information on the robots and the software that we used is available in Annex B.

This chapter is based on the work presented in Degallier et al (2006), Degallier et al (2008) and Degallier et al (2010) (submitted).

5.1 First Implementation on the HOAP-2

Based on the control architecture presented in Chapter 4, a controller for drumming was implemented for the HOAP-2 robot. The set up for the experiment can be seen on Fig. 5.7: the robot sits in front of a drum set composed of three instruments: two bongos and a cymbal. The two arms are controlled, that is four DOFs for each arms (the hand gripper is not controlled as the stick is directly taped to the arm). The big, central bongo can be hit by both arms, the cymbal only by the left one and the small bongo only by the right one. The sticks are made out of compliant hard plastic.

The implementation, depicted on Figure 5.1, consists of four main blocks:

- (i) **Task specification:** The task is defined through a musical score to be played.
- (ii) **Constraints:** The musical score is translated to parameters for the CPGs according to two constraints:
 - (a) *Timing*: when to play a new note of the score (based on the clock)
 - (b) *Target angles*: where to place the arm to reach the drum pads (IK in simulation, look-up table on the real robot)
- (iii) **Whole-body CPG:** The trajectories for each DOF are generated by the CPG.
- (iv) **Position Feedback:** The position of the drum pads is fed back to the system (in simulation).

5.1.1 CPG network and parameters

The CPG network is illustrated on Figure 5.2. We control eight DOFs, four for each arm, plus a clock, that is used as a reference for the general tempo, similarly to a metronome in music. Only one DOF per arm is oscillating, namely the shoulder pitch (L[1]/R[1] on the figure, in green), and it is coupled to the clock in an unilateral way. The other DOFs only produce discrete movements (in blue on the figure). All the DOFs are controlled by

5.1. First Implementation on the HOAP-2

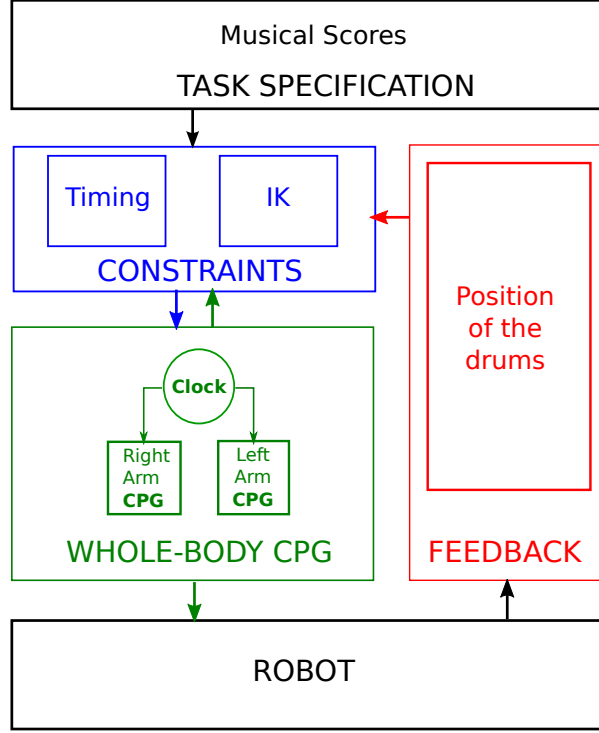


Figure 5.1: First implementation of the drumming behavior. The four DOFs of each arm are controlled by the UPG presented in Chapter 4, all of them are unilaterally coupled to a clock. Scores are transformed into time-varying vectors of parameters for the CPGs according to the constraints of the task (i.e. timing, position of the drums). In simulation, the actual positions of the drums are fed back to the controller which updates the corresponding target angles according to an inverse kinematics algorithm. On the real robot, the target angles are manually defined by placing the arms in an adequate position.

a unit pattern generators defined through the following equations¹ otherwise

$$\dot{y}_i = v_i \quad (5.1)$$

$$\dot{v}_i = -\frac{1}{4}B^2(y_i - \gamma_i) - Bv_i \quad (5.2)$$

$$\dot{x}_i = \frac{A}{|\mu|}(\mu_i - r_i^2)(x_i - y_i) - \omega z_i + n \quad (5.3)$$

$$\dot{z}_i = \frac{A}{|\mu|}(\mu_i - r_i^2)z_i + \omega(x_i - y_i) + n \quad (5.4)$$

where x is the output of the system, y , v and z are auxiliary variables and $r_i = \sqrt{(x_i - y_i)^2 + z_i^2}$. A and B are constant parameters, controlling the rate of convergence

¹Note that this UPG is a previous version of the one that was presented in Chapter 4, Section 4.4.

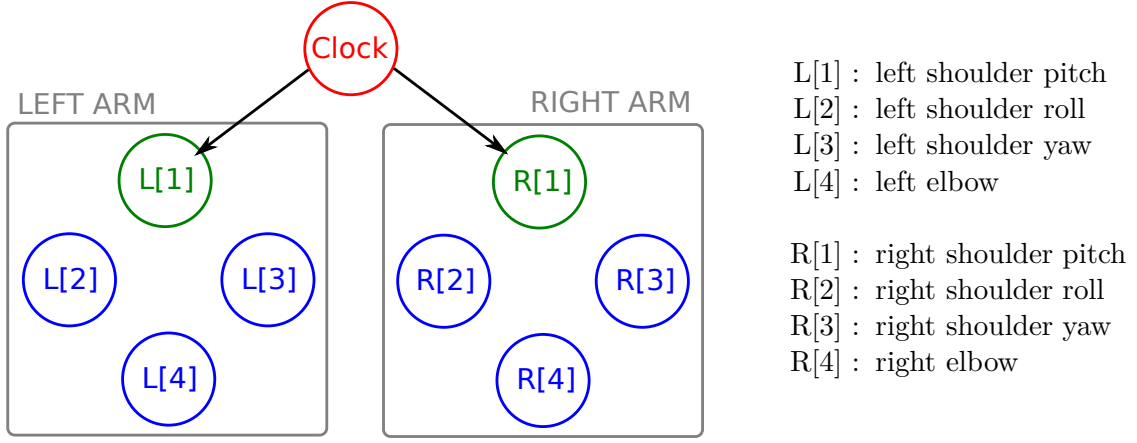


Figure 5.2: CPGs network for drumming for the HOAP-2. Blue circles denote discrete DOFs ($\mu < 0, \forall t$), red circles rhythmic ones ($\gamma = \text{cst}, \forall t$) and green circles both discrete and rhythmic ones. Black arrows denote (active) couplings. See text for discussion.

to the limit cycle and the fixed point respectively, and they are set to the same value for each joint ($B = 20$ and $A = 100$). n is a noise following a normal distribution ($\mathcal{N}(0, 1)$). The output of the system is a sine signal with an offset of γ_i , a frequency ω , and an amplitude $\sqrt{[\mu_i]^+}$, with $[x]^+ = \max(0, x)$. Note that for this particular task, the frequency (ω) is the same for all the joints. The dynamics of the clock is simply given by a simple Hopf oscillator of unit amplitude:

$$\dot{x}_i = A(1 - r_c^2)x_c - \omega z_c \quad (5.5)$$

$$\dot{z}_i = A(1 - r_c^2)z_c + \omega x_c \quad (5.6)$$

where $r_c = \sqrt{(x_c - y_c)^2 + z_c^2}$. In the application, only the shoulder pitch (L[1] / R[1] on the figure) of each arm is oscillating, the output of the other DOFs being always purely discrete. The shoulder pitch of both arms are coupled to the clock in the following way:

$$\dot{x}_i = \frac{A}{|\mu|} (\mu_i - r_i^2) (x_i - y_i) - \omega z_i + K_c x_c + n \quad (5.7)$$

$$\dot{z}_i = \frac{A}{|\mu|} (\mu_i - r_i^2) z_i + \omega (x_i - y_i) + K_c z_c + n \quad (5.8)$$

for $i=L[1], R[1]$, which ensures that they will be in phase. The gain of the coupling was set to $K_c = 4$ to ensure fast convergence. The coupling is unilateral so that perturbations occurring at the level of the arm do not affect the clock. Note that the role of the clock is to compensate for the perturbations of the phase that occur during Hopf bifurcations, i.e. to prevent the phase of the system from changing while a musical score is played.

5.1. First Implementation on the HOAP-2



Figure 5.3: Scores for drumming. Two drumming scores that we make the HOAP-2 robot play. The two lines of notes correspond to the two different hands (upper is left arm and lower right arm) and the height of the note indicates the instrument that has to be played. Here the upper one is the cymbal, the middle one the central bongo and the lower one the small bongo. Score A is thus simply a score where each arm has to beat alternatively their respective instruments. The lower score corresponds to the trajectories shown on Fig.5.5.

5.1.2 Task definition and constraints

The musical scores are encoded into binary matrices (M) of dimension $4 \times n$ where n is the number of notes. The lines correspond to the four possible tasks:

- (1) Small bongo with right arm
- (2) Central bongo with right arm
- (3) Central bongo with left arm
- (4) Cymbal with left arm

If a particular element M_{ij} is equal to 1, it means that the task i has to be performed at time j and otherwise it is equal to 0. Tasks (1) and (2) ((3) and (4) respectively) cannot be active at the same time (one arm cannot beat two drum pads at the same time), but they can simultaneously be null, in which case the arm is said to be idle.

The score matrix has to be transformed into time-varying parameters ω (which is the same for all DOFs), $\vec{\mu}$ and $\vec{\gamma}$ for the CPGs. The frequency of the CPG and the amplitude for the oscillating DOFs can be arbitrarily chosen. They were set to $\omega = \pi$ rad/s (0.5 Hz) and $\mu_i = 0.6$ deg if $i = L[1], L[2]$ and else $\mu_i = -15$ (for purely discrete DOFs) in our case.

Concerning the offsets γ_i , they are constrained by the position of the drums and by the amplitude. To define adequate angles for the two drums, the robot is manually set in an appropriate posture for each instrument (slightly above the drums), then the encoders are read and the values stored in a look-up table. Note that this step has to be done only once at the beginning of the experiment. In our case slight errors in the positioning will be compensated by the compliance of the sticks (e.g. if the target trajectory “goes below” the surface of the drum pad). Table 5.4 summarize the different parameters that we used. We will see in Subsection 5.1.4 how a feedback term on the position of the drums combined with an inverse kinematics algorithm were implemented in simulation to directly obtain appropriate angles.

Left arm	γ_{CB}	γ_{CL}	μ	Right arm	γ_{CB}	γ_{SB}	μ
L[1]	-63.5	-54.5	5.7/-15	R[1]	63.5	59.6	5.7/-15
L[2]	11.5	23.5	-15	R[2]	-11.5	-9.7	-15
L[3]	-14.9	12.6	-15	R[3]	14.9	-19.5	-15
L[4]	-59.6	75.1	-15	R[4]	59.6	55.0	-15

Figure 5.4: Control parameters for drumming. This table summarizes the values for μ and γ for the drumming set-up. The γ were found manually and are indicated in degrees. μ was set to -15 for the non-oscillating joints. The value of μ for the shoulder pitch is either 5.7 degrees (0.1 radians) or -15 depending on whether the joint is oscillating or not. The value -15 was chosen to ensure a fast transition to the non-oscillating state.

Note that if an arm is idle at a given time, its posture will not be specified directly by the score (or, more precisely, by the target instrument). To overcome this, we anticipate the movement in the sense that the arm is set in the position corresponding to the next instrument to be hit in the score, as illustrated on Fig. 5.5.

To determine when to play a new note, the phase of the clock is monitored: the duration of each note is given by the period of the clock (i.e. 2s in our case). The moment where the new note has to be played (i.e. where the CPGs parameters are changed) is chosen during the ascending phase of the movement. More precisely, it was set just after the drum has been hit to maximize the time between the change of parameter and the contact with the drums.

5.1.3 Score playing with the HOAP-2

With this implementation, the robot can play any score at the given frequency of 2 Hz². Movies of the robot drumming the two different scores can be found at www.biorob2.epfl.ch/sd_movies, Movie 1 and 2, and snapshots of the robot drumming are shown in Fig. 5.7.

Fig. 5.5 shows typical trajectories for two joints of the right arm when playing a score (in this case, the Score B of Fig. 5.3). The score is indicated on the background of the figures by colored stripes: a pink stripe means that the central bongo has to be played and a blue one the small bongo (a white background means that the arm is idle). The black curve (that corresponds to the elbow joint, L[4]) illustrates the behavior of the system for purely discrete trajectories. As explained in the previous section, when no drum has to be played (idle case), the target position is set to the one corresponding to the next instrument to be played. The target positions are the ones specified in Table 5.4. The green curve shows the trajectory for the shoulder pitch (L[1]): it is oscillating when a drum has to be beaten ($\mu = 5.7$) and purely discrete when the arm is idle ($\mu = -15$).

²We could go up to 1 Hz but we have chosen to limit the frequency to 0.5 Hz to avoid wearing out the motors.

5.1. First Implementation on the HOAP-2

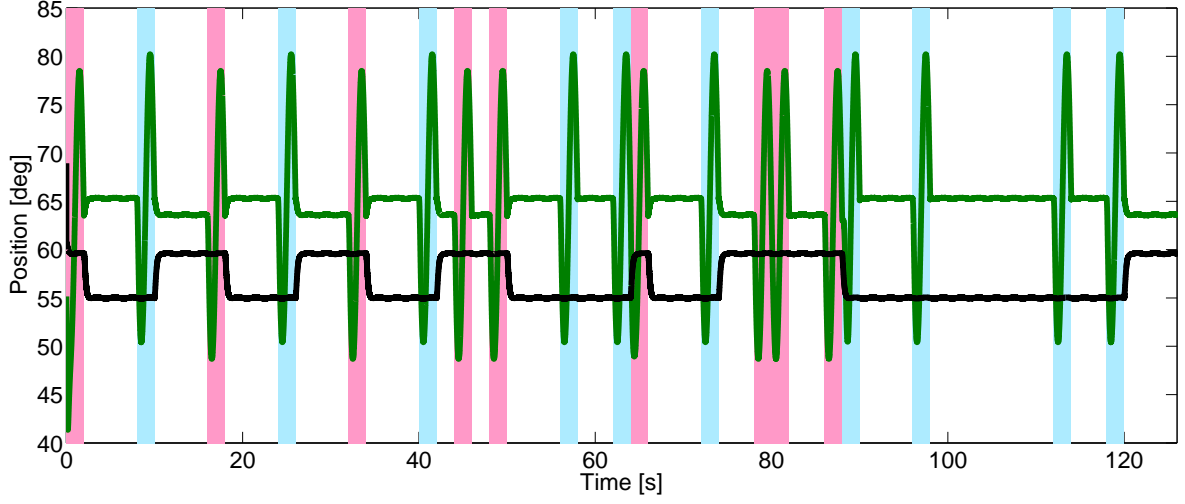


Figure 5.5: Discrete and rhythmic trajectories and score. This figure shows both the score (Score B of Fig. 5.3) that the robot has to play and the absolute values of the trajectories generated by the controller for two joints, the right shoulder pitch (green) and the right elbow pitch (in black). The other joints of the right arm behave analogously to the elbow. The stripes in the background indicate the score, i.e. which drum pad has to be played. Pink corresponds to the central bongo, blue to the small one and no color means that the arm is idle. The pitch joint is oriented such that the arm touches the drum when the joint is at the maximum value. See text for discussion.

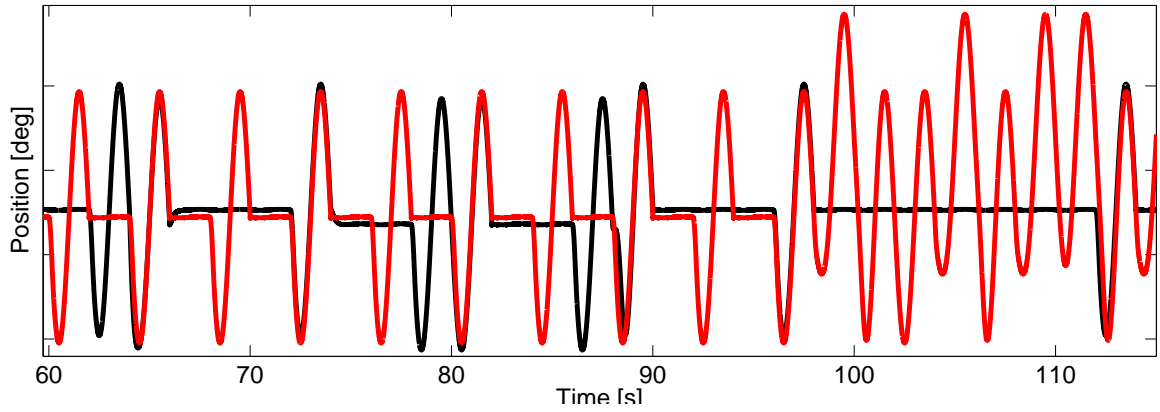


Figure 5.6: Coupling for Score B. This figure shows the trajectory for the shoulder pitch of both arms (left in red and right in black) for a section of Score B. Note that since the joints do no rotate in the same direction we had to take the absolute value. In addition the trajectories were translated to be partly superimposed.

The amplitude of the oscillations is larger than the one that we had specified. It is due to the fact that we have a coupling with a very high gain. This phenomenon could be simply avoided by using dissipative couplings (as presented in Chapter 4), Section 4.5. Note that only the target trajectories are shown on the figure since the difference between the actual and the target trajectories is quite small (less than 1 degree). The maximum error occurs when the drums hit the drum pad. Figure 5.6 shows that the two joints stay synchronized throughout the score thanks to the coupling to the clock. Indeed, without this coupling, the different Hopf bifurcations would lead to perturbations of the phase and thus to loss of synchronization.

It can be seen from the graph and from Movies 1 and 2 (www.biorob2.epfl.ch/sd_movies) that the acceleration is very high at the on-set of a discrete movement. This is due to the fact that, in this first implementation, we are not using a go command to ensure that the desired velocity is null at the beginning of the movement [as explained in Chapter 4, Sect. 4.2].

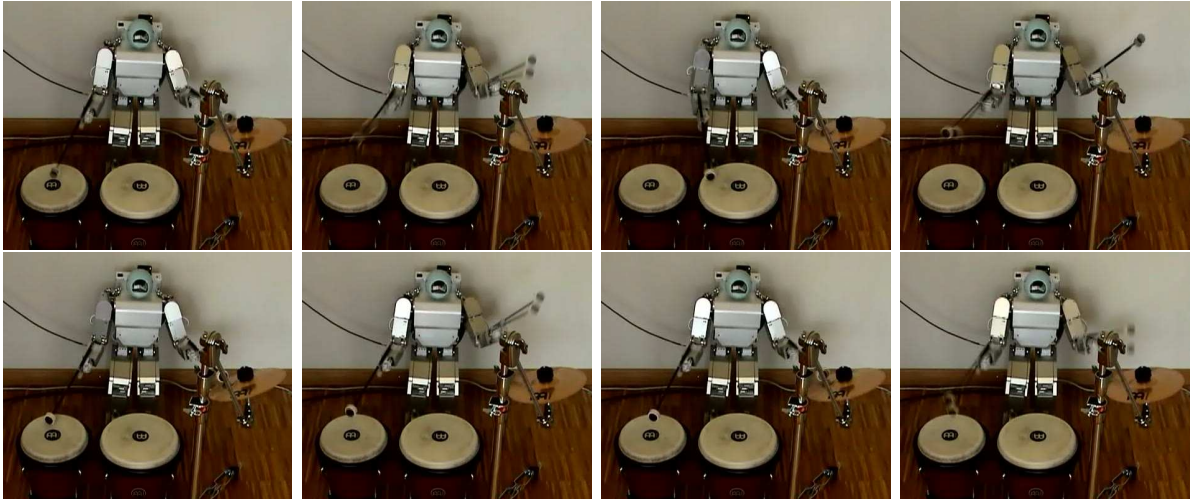


Figure 5.7: Snapshots of the HOAP-2 drumming. The drum set is composed of three instruments; the small bongo (on the left of the figure), the central bongo (in the middle) and the cymbal. The robot is playing a score at the frequency of 0.5 Hz. The frames were taken at regular intervals (0.28 s). Movies of the robot drumming are available at www.biorob2.epfl.ch/sd_movies, Movie 1 and 2.

5.1.4 Drum pads position feedback

In simulation, we take advantage of the fact that we have a perfect knowledge of the environment, and more particularly of the position of the drums, to improve the controller. The position of the drums are constantly updated and transformed into target joint angles based on an inverse kinematics (IK) algorithm. The IK only uses three degrees of

5.1. First Implementation on the HOAP-2

freedom (the angle of the shoulder yaw joint being fixed) so that an analytical solution can be obtained. The target angles of the look-up tables are constantly updated and the robot can thus adapt to changing environment. Fig. 5.8 shows snapshots of the HOAP-2 robots playing a drum pad that is constantly moved.

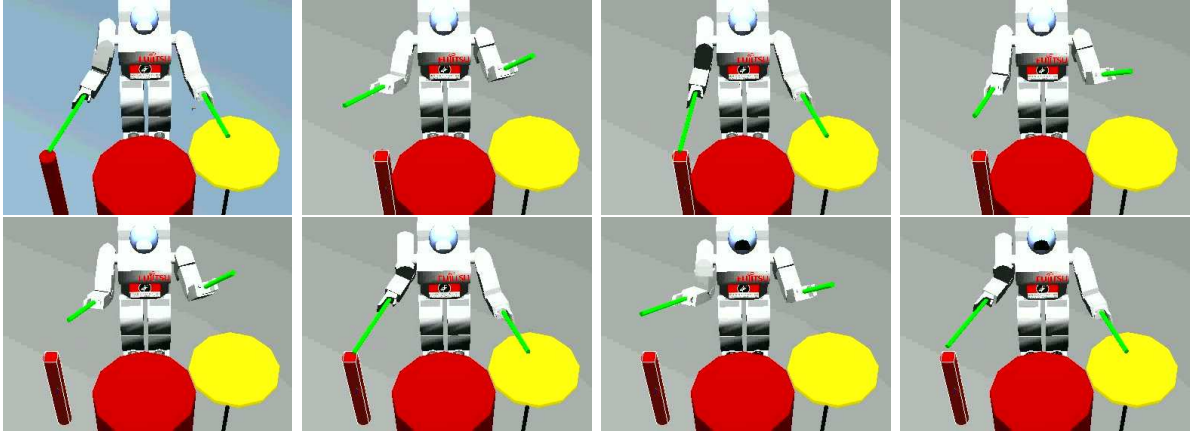


Figure 5.8: Moving drum pads. The trajectories of the arm are modulated according to the constraints given by the drum pad position. The frames were taken at regular intervals (0.56 s). The original movie is available at www.biorob2.epfl.ch/sd_movies, Movie 3.

5.1.5 Concluding remarks on the first implementation

This first application shows that the architecture can easily be applied to the control of robots with multiple degrees of freedom and fast control loops (1 ms). Moreover, the position feedback that is implemented in simulation gives an illustration of coupling between the robot and its environment. A small issue is that the acceleration at the beginning of the discrete movements was large due to the non bell-shaped velocity profile of the speed (since we did not use a go command to smoothen the movement, as explained in Chapter 4).

For the second implementation, we made several enhancements:

- Use the four limbs and a complete electronic drum set;
- Add a GUI to allow the user to play interactively with the robot by defining the score on the fly;
- Improve the feedback on the drum positions by using vision;
- Add a contact feedback to better handle the collision with the drums (instead of using compliant sticks);

- Modify the discrete system to have a bell-shaped velocity profile.

Drumming was one of the very first demonstrations to be run on the newly developed robot. Some of the current features of the robot were not implemented at that time. In particular the force sensors were not available, so that we had to use the information coming from the electronic drums instead to implement the contact feedback. In addition we had to use velocity control instead of position control since the position mode of the iCub is designed to follow a minimum jerk trajectory.

5.2 Second Implementation on the iCub

The general set up for the second implementation can be seen on the snapshots in Fig. 5.12: the robot is fixed to a metallic structure by the hips and plays on an electronic drum set. The four limbs together with the head are controlled. We control actively four joints for each limb (the shoulder pitch, roll and yaw and the elbow for the arms, and the hips pitch, roll and yaw and the knee for the legs) and the first three DOFs of the head (neck pitch, neck roll and neck yaw). The wooden sticks are grasped by the hands which remain closed afterwards. The pedals are placed so that the robot can easily reach them when its legs are stretched.

The implementation, depicted on Figure 5.9, is similar to the first one. It consists of same four blocks:

- (i) **Task specification:** a graphical interface (GUI) that allows a user to define the behavior (i.e. the musical score) of the robot online.
- (ii) **Constraints:** the control commands are adapted to the actual environment and state of the system; this is done through two subsystems:
 - (a) *Timing:* when to play a new note of the score (based on the clock)
 - (b) *Target angles:* where to place the arm to reach the drum pads (IK in simulation, look-up table on the real robot)
- (iii) **Whole-body CPG:** The trajectories for each DOF are generated by the CPG.
- (iv) **Feedback:** the trajectories are modulated by two feedback loops
 - (a) *Contact:* The movement is stopped when a collision with an obstacle is detected.
 - (b) *Visual tracker:* Vision is used to track and update the Cartesian position of the drums (in simulation).

5.2. Second Implementation on the iCub

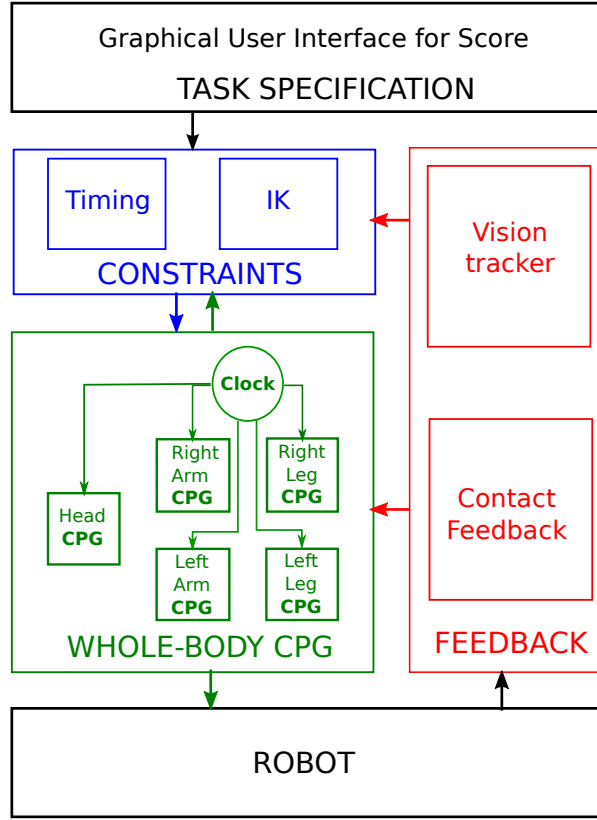


Figure 5.9: Second implementation of the drumming behavior. This implementation is designed so that any user can interact with the robot to make it play a score of his/her choice through a simple graphical user interface (see Fig. 5.11). Visual feedback is added to the system so that the robot can detect markers placed on drums and autonomously adapt its movement to their position. Finally, a feedback to deal with collisions between the arms of the robot and the drums is added for safety reasons. Five parts are controlled, namely the head, the left arm, the right arm, the left leg and the right leg. Green arrows denote couplings.

We have to control the robot in velocity (rather than in position), because the low-level position controller is designed to produce movements with zero initial and end velocities, which may lead to interruptions in our case (as we are sending new commands every 15 ms). The CPG states are used to compute both the desired states and the desired velocities for each joints. These values are then transformed into velocity command through a PID controller.

5.2.1 Design of the whole-body CPG

The CPG network is illustrated on Figure 5.10. Five parts are controlled, namely the left and right arms, the left and right legs, and the head. For the four limbs, we control only

the first four joints, the other ones being locked. Similarly to the previous implementation a clock is used as a absolute reference time to avoid phase shifting due to the Hopf bifurcations. In both the arms and the legs, two joints are oscillating, the shoulder/hip pitch (RA[1], LA[1], RL[1], LL[1]) and the elbow/knee (RA[4], LA[4], RL[4], LL[4]). They are all unilaterally coupled to the clock, but contrarily to the previous implementation the phase shift is no longer constant and can be modulated by the user on the fly. The head has two oscillating joints; the head pitch (H[1]) and the head yaw (H[2]). They are not meant to oscillate simultaneously and thus are only coupled to the clock. The head yaw is used to scan the entire drum set to detect the position of the drum pads, while the head pitch oscillating movement is used for aesthetic reasons (to get a behavior that looks more human-like).

We add an equation for the go command to ensure that the velocity is null at the onset of the movement and the system becomes:

$$\dot{h}_i = 1 - h_i \quad (5.9)$$

$$\dot{y}_i = v_i \quad (5.10)$$

$$\dot{v}_i = -\frac{1}{4}B^2h_i^2(y_i - \gamma_i) - Bh_iv_i \quad (5.11)$$

$$\dot{x}_i = \frac{A}{|\mu_i|} (m_i - r_i^2) (x_i - y_i) - \omega_i z_i + n \quad (5.12)$$

$$\dot{z}_i = \frac{A}{|\mu_i|} (m_i - r_i^2) z_i + \omega_i (x_i - y_i) + n \quad (5.13)$$

with $B = 5$ and $A = 2$. The system for the clock is the same as for the first implementation (Eqs. 5.5-5.6). The phase shift between the different parts (θ_P) and the clock is now a parameter and the coupling term is given by

$$\dot{x}_i = \dots + K_c(\cos(\theta_P)x_c - \sin(\theta_P)z_c) \quad (5.14)$$

$$\dot{z}_i = \dots + K_c(\sin(\theta_P)x_c + \cos(\theta_P)z_c) \quad (5.15)$$

for $i = H[1], H[2], RA[1], LA[1], RL[1], LL[1]$ and where we set $K_c = 2$. Note that the phase shift is the same for all the joints of a particular limb and that is why it is denoted by the index P (instead of i).

5.2.2 Task specification and Constraints

The controller was developed so that any user can define the score that the robot is playing on the fly. A graphical interface (based on the open source library Qt3), shown on Fig. 5.11, was developed to ease the control of the robot. The open parameters are the following:

- for each arm: ID of the target drum or idle, phase shift relatively to the clock

5.2. Second Implementation on the iCub

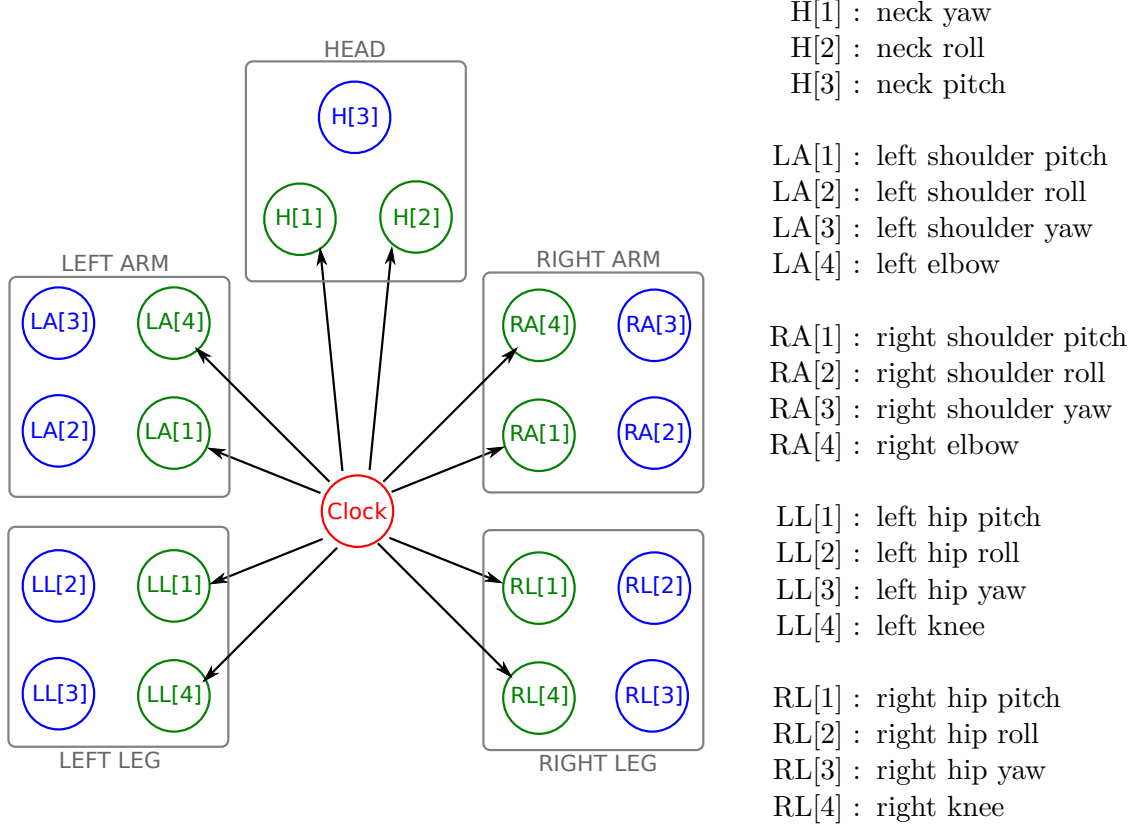


Figure 5.10: CPG network for drumming for the iCub. Blue circles denote discrete DOFs ($\mu < 0, \forall t$), red circles rhythmic ones ($\gamma = \text{cst}, \forall t$) and green circles both discrete and rhythmic ones. Black arrows denote (active) couplings.

- for each leg: drumming or idle, phase shift relatively to the clock
- for the head: idle or scanning (to locate drums) or looking at one of the drums
- for the whole system: the frequency (an upper bound of 1 Hz was fixed)

The user can modify the score at any time, however the parameters of the CPG are changed only when it is safe for the robot, that is during the period where the limb moves away from the drums. An intermediate module, called the task manager, is thus responsible to monitor the phase of each limb and to send the new commands during the secure phase. The frequency, the amplitude and the phase shifts can be sent to the CPGs without transformation. As for the drums ID, they are mapped to target joint angles (for each controlled DOF of the limb) defined relatively to the position of the drums. These target joint angles can be either predefined or determined through a visual tracker system combined with an inverse kinematics algorithm, as will be discussed below.

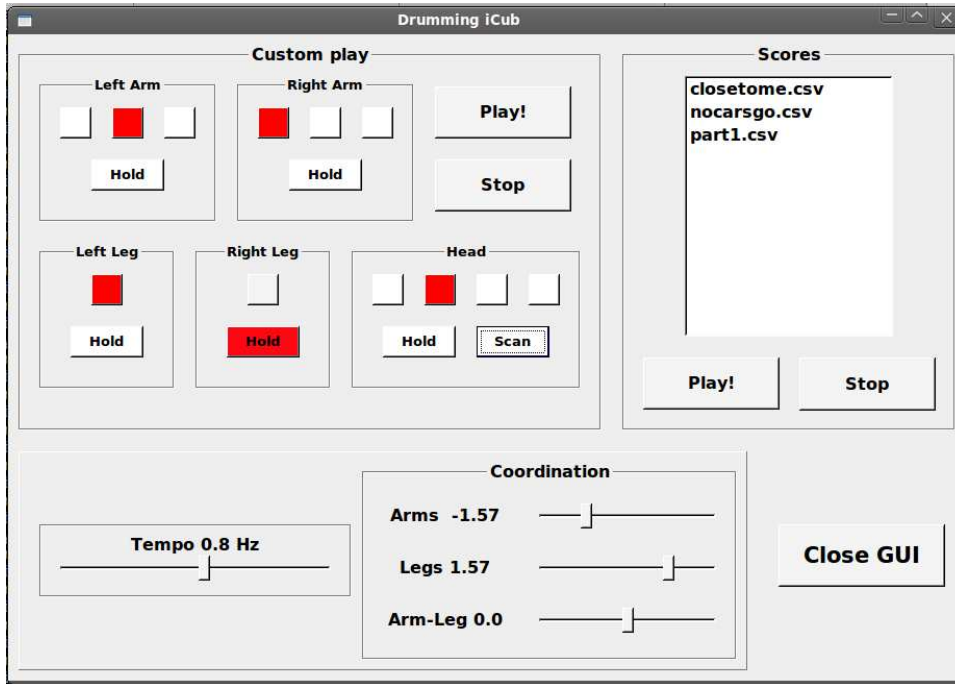


Figure 5.11: Graphical user interface for drumming. The user can choose between two modes: custom play, where (s)he can choose online what each limb of the robot is doing, or a predefined score. (S)he can also modify the phase shift between the limbs and the general frequency at which the robot is playing.

5.2.3 Interactive drumming with the iCub

With this implementation, the robot can perform any score defined with the GUI in a robust way: we were able to run the demonstration for hours with random users modulating on the fly the score that the robot was playing. Snapshots of the robot playing the drums are shown on Figs. 5.12 and movies are available at biorob2.epfl.ch/sd_movies, Movie 4.

Figs. 5.13 illustrates the trajectories obtained for drumming in open loop on the iCub robot. Note that the robot touch the drums when the trajectory is at its maximum due to direction of rotation of the motor. Fig. 5.13 shows how a UPG can be modulated through the three parameters corresponding to the offset, the amplitude and the frequency. The convergence between the reception of the new commands by the CPG (indicated by the blue vertical lines) and the desired state is very fast. Both the desired (black, dashed line) and the actual (red line) trajectories are shown on the figure: if in general the tracking is very good, it can be seen that at $t \approx 80s$ the tracking is lost and the robot can not follow the desired trajectory. The gains chosen for the PID of the velocity command being quite low, using higher values may solve this issue. The control of the phase shift between the

5.2. Second Implementation on the iCub

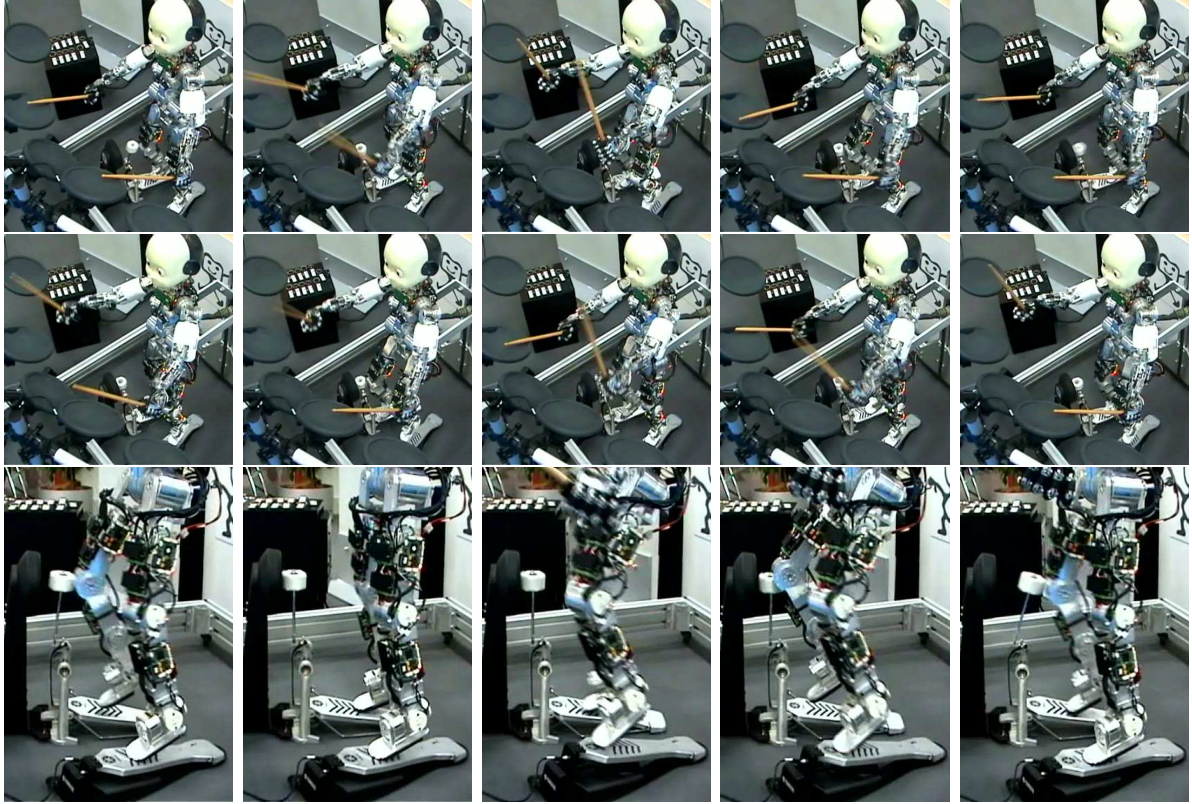


Figure 5.12: Snapshots of the iCub drumming (Automatica fair, Munich, 2008). The robot is playing with , the score being defined online by a user through a graphical interface. Note that the robot is kept in a upright posture through a metallic structure. Movie available at http://biorob2.epfl.ch/sd_movies, Movie 4. The frames were taken at regular intervals (0.45 s).

two hips is shown on Fig. 5.14. It can be seen that the transitions from one configuration to another happen in less than half a cycle and that they are smooth.

5.2.4 Contact Feedback

Since we control the robot in velocity (rather than in force), and since the motors are not compliant (yet), a feedback policy to safely and smoothly handle collisions with the drums was added to deal with modeling imprecisions. Such feedback allows for a safe interaction with the environment, as it avoids high strains in the joints when the movement is prevented by an obstacle (e.g. a drum pad in our case).

The contact feedback term (Ijspeert et al (2002)) is designed so that the robot stops its movement in the current position when an obstacle is detected, and resumes to the normal trajectory when it is safe again. Collision detection is made through the electronic

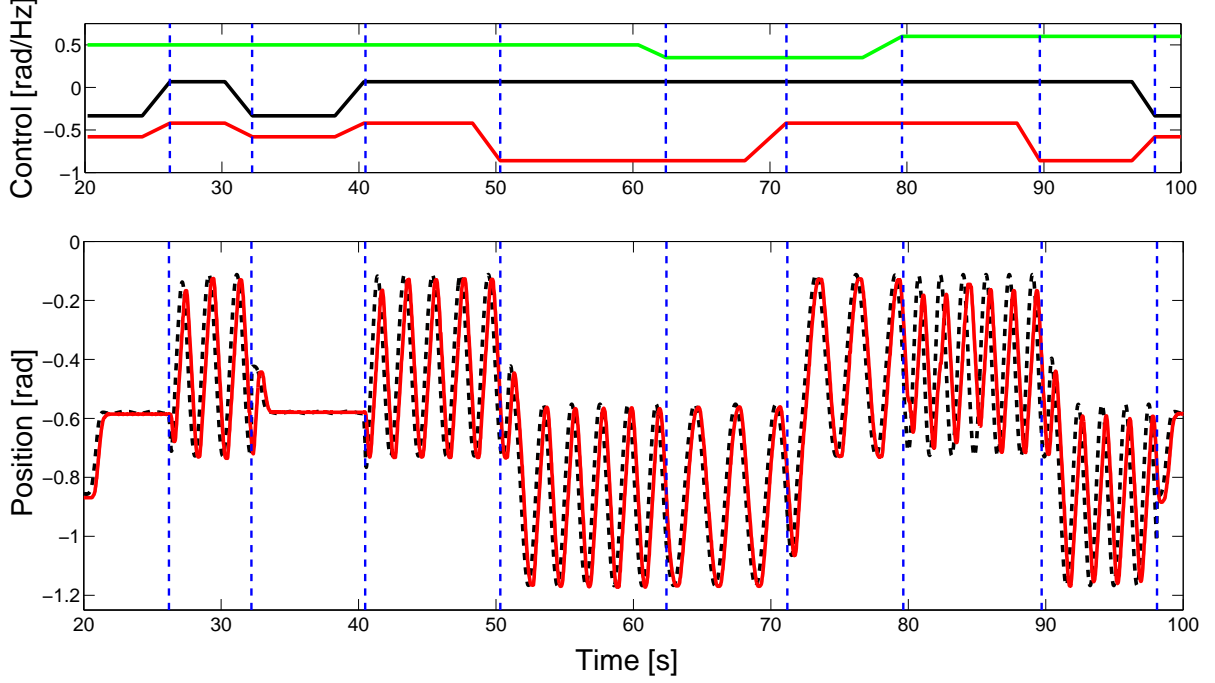


Figure 5.13: Control commands for a DOF. The figure shows both the actual (in red) and the target (in black) trajectories for the left shoulder pitch joint (top panel) and the corresponding control commands (bottom panel). The blue vertical lines indicate when the new control commands are received by the CPG.

drum set (as no feedback from the robot was available at the time). A message is sent to the controller indicating which drum had been hit and the current (desired) position of the corresponding limb is stored as the reference attractive point (\hat{x}).

The system of equations for the UPGs is modified in the following way: an attractor with a high gain is activated to stop the movement in its current position \hat{x}_i (in Eq. 5.16) together with a term that slows down the trajectory (in Eq. 5.17), i.e. we have

$$\dot{x} = a(m_i - r_i^2)(x_i - y_i) - \omega z_i + \alpha_x(\hat{x}_i - x_i); \quad (5.16)$$

$$\dot{z} = \frac{a(m_i - r_i^2)z_i + \omega(x_i - y_i)}{1 + \alpha_y(\hat{x}_i - x_i)^2} \quad (5.17)$$

where \hat{x}_i is the current position of joint i when the feedback is received. The new term in the first equation corresponds to the simplest expression for an attractor, with α_x controlling the gain of convergence. In the second equation, the denominator is a term that decreases the value of \dot{z} , and thus slows down the movement. The square of the difference between the reference attractive point and the current position is taken here, so that the effect on the speed is proportional to the error and always positive. Note

5.2. Second Implementation on the iCub

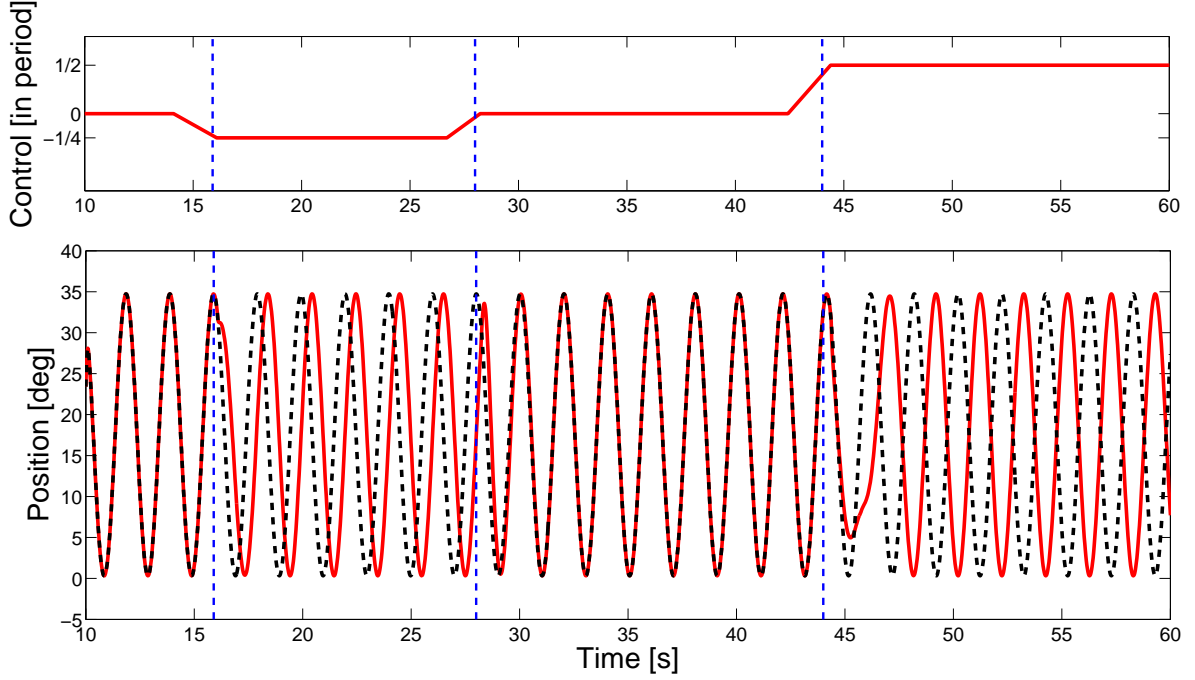


Figure 5.14: Control of the phase shift. The figure shows the trajectories of the right (in red) and left (in black) hip pitch (bottom panel) and the corresponding phase shift (top panel). The blue vertical lines indicate when the new control commands are received by the limb CPG.

that the absolute value of the difference would lead to similar results. The integer one is added to the expression so that the system is the same as the open loop one when the system has converged (i.e., $x_i = \hat{x}_i$). The parameters α_x and α_y are set to high values when a collision is detected and to zero otherwise. The exact values of α_x and α_y depends on the characteristics of the robot (time step, motor specifications, ...) and have been chosen by hand in our case.

Note that to resume to the normal trajectory we need to know when the feedback is not anymore necessary, i.e. when the trajectory moves away from the obstacle. In order to do that a so-called observer (i.e. a copy of the CPG that is unaffected by the feedback) was used to monitor the direction of the unperturbed trajectory.

Figure 5.16 shows snapshots of the robot adapting its trajectory to different positions of the drum pad and a movies is available at biorob2.epfl.ch/sd_movies, Movie 6. Figure 5.15 illustrates the effect of the feedback in real application where the robot is drumming with a relatively high frequency (1 Hz). First the robot is in a rest position, then at $t \approx 5s$, it starts drumming. During the first cycle, no contact with the drums occur, then, at $t \approx 1.2s$, a collision is detected and the arm is stopped in its current position. At $t \approx 1.7s$, the normal trajectory being safe again, the feedback is removed

and the arm start moving again. For the next two cycles, the drum pad is moved in different positions. It can be seen that the efficiency of the feedback is related to the speed at the time of impact: the higher the speed, the longer it takes to the system to stabilize to the fixed point. Thus the feedback will be more efficient if the impact occurs near the peaks of amplitude (*middle case on the figure*). However, even when a collision occurs close to the peak of velocity (*right case*) the feedback successfully stabilizes the arm in its current position.

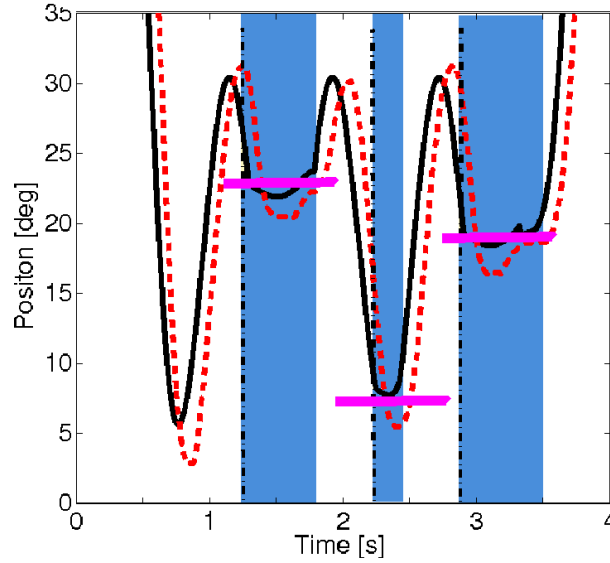


Figure 5.15: Contact feedback: trajectories. Right shoulder pitch trajectories while drumming. The colored areas correspond to the interval of time where the feedback is active, the plain line to the desired trajectory and the dashed one to the actual trajectory. The horizontal thick lines denote the approximated position of the drums. Snapshots of the behavior of the system with contact feedback is depicted in Fig.5.16. Here $\alpha_x = 100$ and $\alpha_y = 100$.

5.2.5 Visual feedback

We have seen that in the first implementation of the drumming, the position of the drums were fed back to the system so that the target position can automatically be derived for the arms through an inverse kinematics algorithm. A similar system was developed for the iCub, but this time based on vision, since the real robot has cameras. For the inverse kinematics, we used the library iKin that is provided with the iCub (see Annex B) .

The visual tracker, that was developed in our lab by Sebastien Gay (Gay et al (2010)), uses the open source ARToolkitPlus (Wagner and Schmalstieg (2007)) to compute the 3D position and orientation of markers placed near the drums. The actual position of the center of each drum is then computed according to a predefined 3D offset between the

5.2. Second Implementation on the iCub

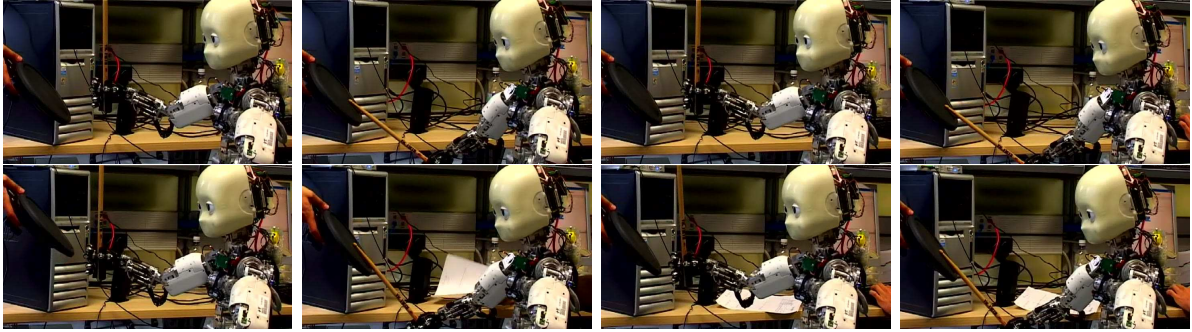


Figure 5.16: Snapshots of the robot drumming with contact feedback. The drum pad is moved while the robot is drumming (between first and second snapshots). The movement of the arm is adapted to the new situation: it is stopped as soon as it touches the drum pad. Note that the drum pad could not be held in this position without the feedback, as the robot is controlled in position, with a high gear motors. Movie available at http://biorob2.epfl.ch/sd_movies, Movie 5. The frames were taken at regular intervals (0.84 s).

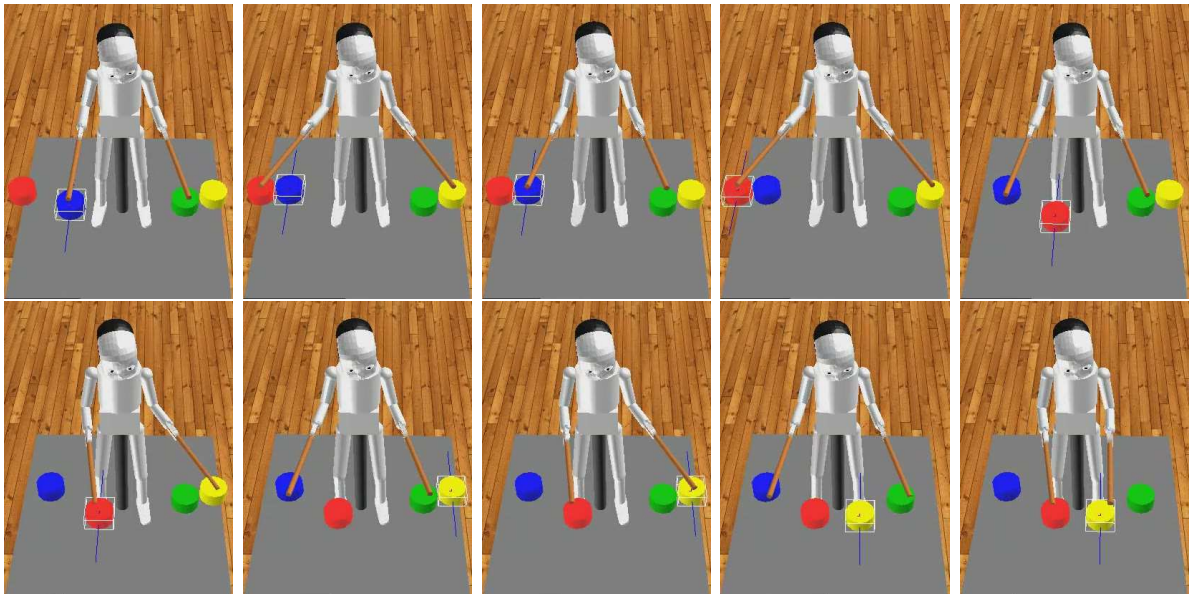


Figure 5.17: Simulation of drumming with moving drums. The target angles corresponding to the different drums are updated continuously so that the robot can adapt its trajectories to their actual position. Here the robot beats alternatively the two drum pads corresponding to each arm. Every new beat the position of one of the drums is changed. Movie available at http://biorob2.epfl.ch/sd_movies, Movie 6. The frames were taken at regular intervals (0.45 s).

marker and the drum. The corresponding target positions for each DOFs (γ_i) of the closest arm of the robot are computed using an extended version of the default inverse kinematics solver provided by iKin³. Note that since that the inverse kinematics is solved through optimization, we are not sure to converge to a solution and the maximal error was set to 5 cm to ensure that suitable angles can be found rapidly enough. The updated target positions are then sent to the controller and are immediately updated.

Thanks to this feedback, the robot can autonomously adapt to time-varying configuration of the drums and also to moving drums. Note that for now this feedback loop has only been fully implemented in simulation, but it will be implemented on the real robot in the future. Fig. 5.17 shows snapshots of the robot drumming on moving pads and the corresponding movie can be found at biorob2.epfl.ch/sd_movies, Movie 6.

5.2.6 Concluding remarks on the second application

This second implementation of drumming illustrates both the adaptivity and the robustness of the architecture. Indeed, once the CPG has been appropriately designed, the robot can perform any score (up to the frequency upper limit) in a robust way. In addition, thanks to the contact feedback, interactions with the drum pads are ensured to be safe for the robot for any frequency. Finally, it has been shown in simulation that simple visual tracking of objects allows for the online adaptation of the movement of the robot in a changing environment, a behavior that will soon be implemented on the real robot.

³Sebastien Gay modified the solver to include the drumming stick as an additional link in the iCub arm chain so that the tip of the stick reaches the center of the drum

CHAPTER 6

APPLICATION TO CRAWLING

Crawling is the first stage of locomotion in infants; it allows them to explore their environments and to move towards persons or objects of interest. This behavior was implemented on the iCub in simulation and partly on the real robot; we developed a controller that allows for modulations of the locomotion, such as changes of speed and steering, and integrates both contact and visual feedback. Contact feedback is used to trigger transitions between swing and stance according to load information in order to increase locomotion stability; visual feedback is used to detect obstacles and objects of interest, and more precisely to create a map of the environment surrounding the robot. Simple reaching for marks on the ground based on vision has also been implemented. Finally, a high-level planner algorithm based on potential fields was combined with the CPG to obtain an autonomous, infant-like behavior (in simulation for now) where the robot, attracted by an object of interest, moves towards it while avoiding obstacles, and finally reaches for it. Steady-state crawling and reaching have been implemented on the real robot iCub.

Note that this section is partly based on work published before in conference articles, notably a previous implementation of crawling based on infant data analysis (Righetti and Ijspeert (2006a)) and a study of the combination of discrete and rhythmic movements for switching between crawling and reaching (Degallier et al (2007)). A synthetic version of this section, excluding visual feedback, reaching and the high-level planning, was presented in Degallier et al (2008) and in Degallier et al (2010) (submitted). We will also briefly present some work done by others to show how they can be integrated to the system: a contact feedback policy developed by Ludovic Righetti and presented in Righetti and Ijspeert (2008) and a navigation planner based on force fields developed by Sebastien Gay and presented in Gay et al (2010).

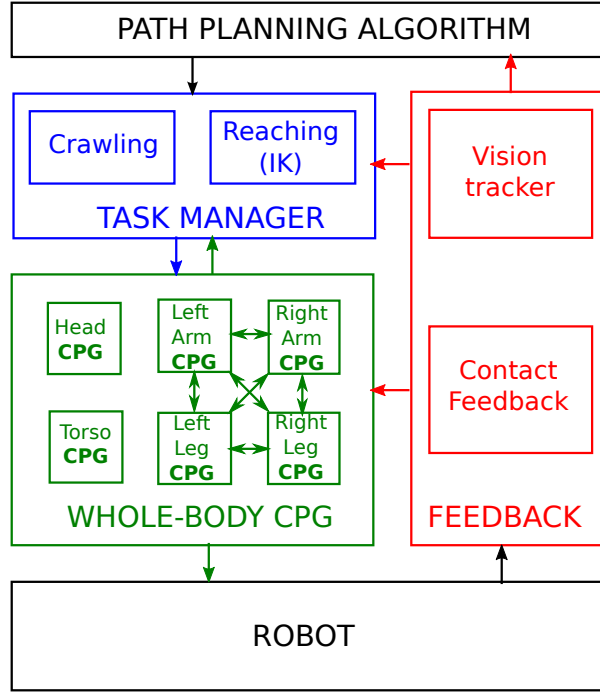


Figure 6.1: Implementation of the crawling behavior. This implementation is designed so that the robot can autonomously evolve in a (time-varying) environment containing obstacles and target objects. It can be seen from the figure that the CPG output is modulated both by high-level and feedback commands. The high-level commands can be either triggered manually by the user (not depicted on the figure) or through visual feedback for reaching (if the robot is close enough to a target object) or for steering the robot to avoid obstacles and move towards target object according to a path-planning algorithm based on potential fields. Obstacles and targets are indicated by ARToolKitPlus markers. There is also a local feedback based on force sensors that modulates the behavior of the robot according to the ground contact information. Six parts are controlled, namely the head, the torso, the left arm, the right arm, the left leg and the right leg. Green arrows denote the couplings between the different parts.

6.1 Software Implementation

The implementation, depicted on Fig. 6.1, consists of four main blocks:

- (i) **High-level planner:** A path avoiding the obstacles and reaching points of interest is determined through an algorithm based on potential fields.
- (ii) **Task manager:** a module that sends the parameters according to the task to be performed, that are here:

6.2. CPG design

- (a) *Crawling*: The manager sends the parameters corresponding to crawling, with the turning angle and the speed of locomotion as open parameters.
- (b) *Reaching (IK)*: When the robot is close enough to a target, it stops and reaches for it; the joint angles for the reaching arm being provided by the iKin library.
- (iii) **Whole-body CPG**: the network consist of a CPG for each of the four limbs and the head and torso; the four limbs are coupled together to obtain a trot gait.
- (iv) **Feedback**: the information from the robot is used to modulate the trajectories; it consists in two subsystems
 - (a) *Contact feedback*: The load information is used to control the transition of each limb between swing and stance (Work by Ludovic Righetti).
 - (b) *Visual tracker*: Vision is used to detect obstacle and object of interest, indicated by predefined marks (Work by Sebastien Gay).

In this application, both arms and legs are controlled as well as the head and the torso. For each arm and leg, we actively control 4 DOFs, that are the shoulders pitch, roll and yaw and the elbows for the arms and the hips pitch, roll and yaw and the knee for the legs; the six degrees of freedom of the head and the three of the torso are also controlled. We thus actively control 22 DOFs. The remaining DOFs are set in particular position at the beginning of the task and remain fixed at that position afterwards.

6.2 CPG design

The design of the controller is based on observations on the crawling of human infants (Righetti (2008); Righetti and Ijspeert (2006a)). While infants can have various locomotion strategies prior to walking, most of them crawl on hands and knees using a gait that is close to a walking trot. The crawling gait has a duty factor higher than 50 % (i.e. the duration of the stance phase is longer than half of a step cycle). The duration of stance is highly correlated with speed of locomotion while the duration of swing remains constant, as is generally observed in quadruped mammals. We explain in the following how such observations were used to construct the crawling controller.

First, in order to be able to set the duty factor and to control the duration of the stance phase and of the swing phase independently, we modified the frequency ω_i of the oscillator described by Eqs. 4.17-4.18 as previously done by Righetti and Ijspeert (2006a). We made it a function of two variables, ω_{swing} and ω_{stance} , that explicitly and independently control the swing and stance durations. ω_i is written as a simple switch between both quantities depending on the state of the oscillator

$$\omega_i = \frac{\omega_{\text{swing}}}{e^{-fz_i} + 1} + \frac{\omega_{\text{stance}}}{e^{fz_i} + 1} \quad (6.1)$$

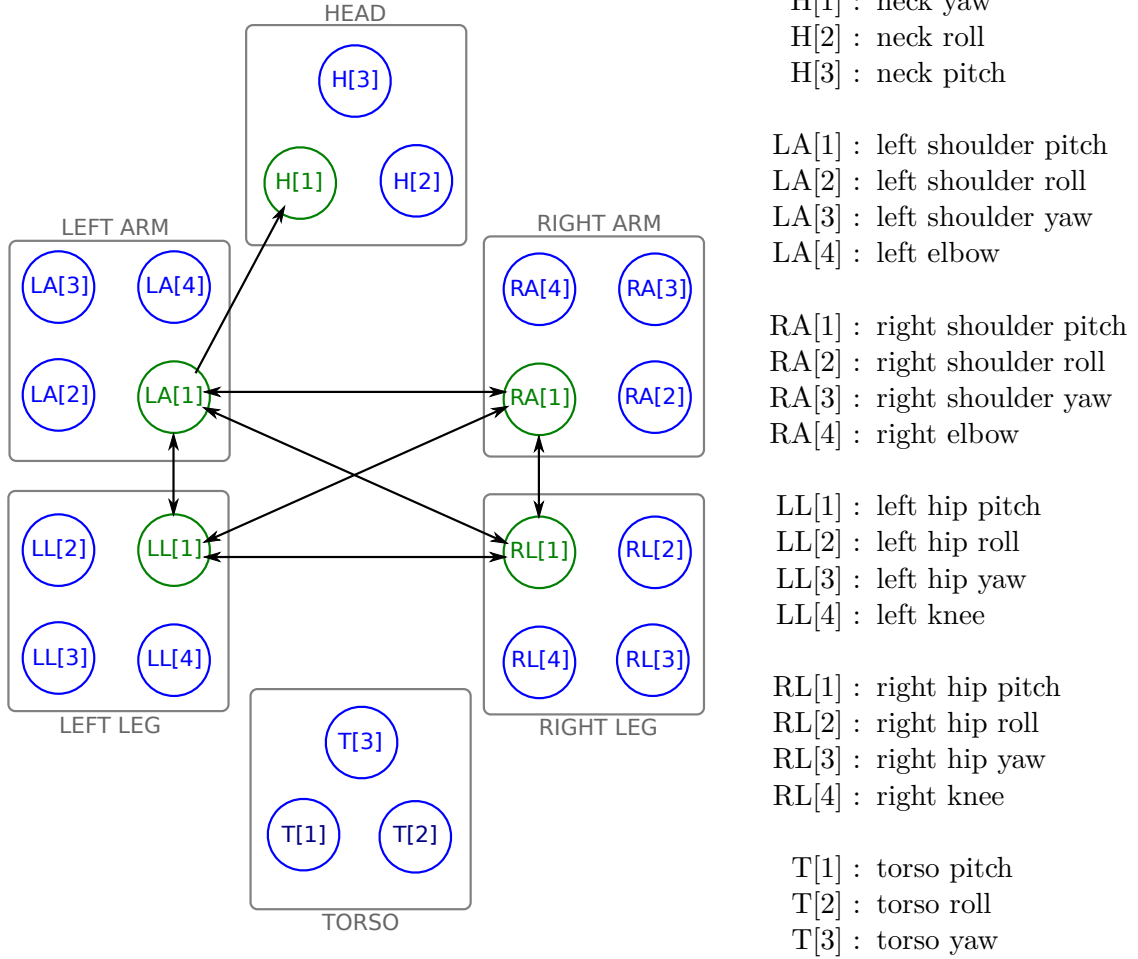


Figure 6.2: CPG network for crawling for the iCub. Blue circles denote discrete DOFs ($\mu < 0, \forall t$), red circles rhythmic ones ($\gamma = \text{cst}, \forall t$) and green circles both discrete and rhythmic ones. Black arrows denote (active) couplings.

where f is a parameter controlling the duration of the switch between the two phases. Only changing the duration of the stance allow us to modify the speed, as illustrated by Movie 9 (biorob2.epfl.ch/sd_movies). Fig. 6.3 illustrates the modulation of the original sine (in red) with a four times longer or four times shorter stance (in blue and in black respectively). In all the trajectories, the general frequency is fixed to $\omega = 4\pi$ and we modulate the duty factor of the walking cycle, defined as $d = T_{\text{stance}} / (T_{\text{stance}} + T_{\text{swing}})$ if $T_i = 1/\omega_i$ for $i = \text{swing, stance}$. The swing period is then computed according to the stance period and the overall period (i.e. $T_{\text{swing}} = 2T - T_{\text{stance}}$). For the red trajectory, $\omega_{\text{swing}} = \omega_{\text{stance}}$ and the resulting trajectory is a symmetric sine (duty factor $d = 0.5$), for

6.2. CPG design

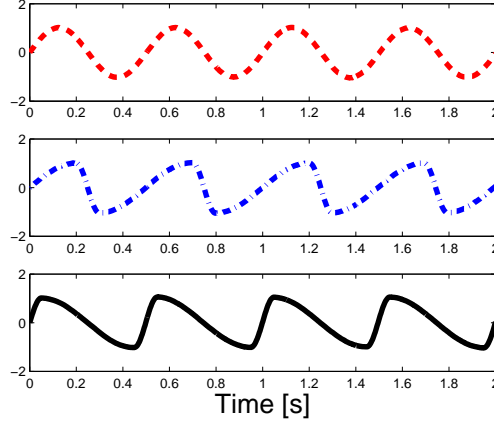


Figure 6.3: Modulation of stance duration. Here $f = 100$, $b = 10$, $a = 5$, $\omega_i = 4\pi$ and the matlab function `randn` to generate the noise, the time step of integration being set to 0.001. See text for discussion. The behavioral results of the change of the stance duration is illustrated by the movie at bioro2.epfl.ch/sd_movies, Movie 9.

the blue curve, $\omega_{\text{stance}} = 2.5\pi$ and $d = 0.8$, i.e. the stance lasts four times longer than the swing, and for the black curve $\omega_{\text{stance}} = 10\pi$ and $d = 0.2$, i.e the stance last four times shorter than the swing.

The CPG network is illustrated in Fig. 6.2. We control the 6 parts of the robot (head, torso, left and right arm, left and right leg). For the four limbs, we control only the first four joints, the other ones being locked. We control the three joint of the torso and the three first joints of the head (i.e. neck pitch, neck roll and neck yaw). Thus we control 14 joints in total.

The head yaw is coupled to the left arm and this rotation is used to enhance the field of vision of the robot for the visual tracker. To ensure the trot gait, the oscillators of shoulders and hips pitch joints are coupled together in the following way

$$\dot{x}_i = \dots + 0 \quad (6.2)$$

$$\dot{z}_i = \dots + \sum_{j \neq i} (\sin(\theta_{ij})x_j + \cos(\theta_{ij})z_j) \quad (6.3)$$

That is, compared to Eqs. 4.19 and 4.20, the k_{ij}^x s are set to 0 and the K_{ij}^z s to 1. Then, the θ_{ij} are chosen as described for the trot in Section 4, Table 4.2.

Note the difference in limb geometry between the arms and the legs coupled with the closed-loop constraints require a precise control of the respective amplitude of the limbs: The specification of the hip pitch amplitude imposes the one of the shoulder pitch due to the close-kinematic loop constraints; the shoulder pitch amplitude is deduced using the forward kinematics provided by iKin.

Concerning the joints other than the hips/shoulders pitches, they are controlled in the

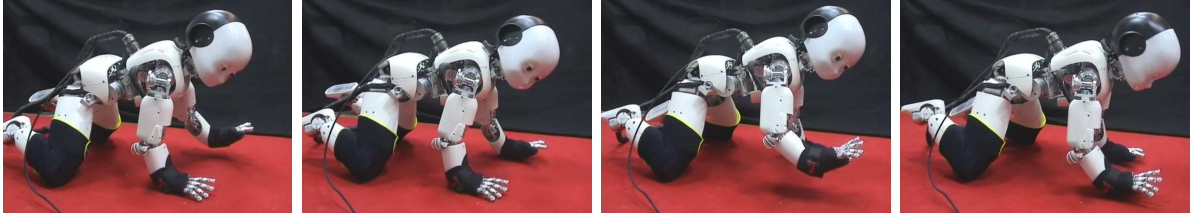


Figure 6.4: Snapshots of the robot crawling. Steady state crawling, with $\omega_{\text{stance}} = \omega_{\text{swing}} = 0.3$. Note that the hands are protected by wrist sport pads because of their fragility and the knees are also covered by sport pads as they are extremely slippery due to their shape and material. The robot turns its head while walking to enhance the detection of the visual markers. Movie available at http://biorob2.epfl.ch/sd_movies, Movie 7.

following way. The shoulder roll, the elbow and the hip roll are kept in a fixed position during the stance and move proportionally to the speed of the shoulder pitch joint during swing to ensure that the knees and the hands are lifted enough to avoid collision with the ground., i.e.

$$g_i = w_i z_j \quad (6.4)$$

where j = shoulder pitch if i = shoulder roll or elbow and j = hip pitch if i = hip roll, where the w_i are chosen by hand. Fig. 6.4 shows snapshots of the robot crawling with the parameters of Table 6.1 with a duty factor of 50% ($\omega_{\text{stance}} = \omega_{\text{swing}} = 0.3$).

6.3 Choice of parameters

We now briefly explain how the parameters of Table 6.1 were chosen. Due to the angle and torque limitations of the robot, the range of possible parameters was relatively small and we selected them by hand. However the performances might be improved by optimizing these parameters.

The initial posture of the robot (that correspond to the position of the limb at the middle of the stance) was chosen so that the elbow is fully extended. Indeed, it has been observed from the simulations that the torque induced by the body weight on the elbow joint is critically high compare to the motor torque available. Thus, the arm position minimizing this torque was chosen (arm fully extended) and the pitch angles of the shoulder and hips were chosen so that the limb is perpendicular to the ground. The field of vision of the robot being restrained by the small angle range of the neck pitch joint, the torso pitch joint was set to the maximal value to increase it.

During the crawling, the closed kinematics chains induced by the contacts with the ground constrain the parameter values for the legs and arms. Since the elbow DOF cannot be moved during the stance because of torque limitations, the pitching and rolling

6.4. Switching between crawling and reaching

left arm	g	A	right arm	g	A
sh pitch	-1.20	0.09	sh pitch	-1.20	0.09
sh roll	0.35	-5.00	sh roll	0.35	-5.00
sh yaw	0.26	-5.00	sh yaw	0.26	-5.00
elbow	0.50	-5.00	elbow	0.50	-5.00
left leg	g	A	right leg	g	A
hip pitch	1.40	0.10	hip pitch	1.40	0.10
hip roll	0.40	-5.00	hip roll	0.40	-5.00
hip yaw	0.00	-5.00	hip yaw	0.00	-5.00
knee	-2.00	-5.00	knee	-2.00	-5.00

Table 6.1: Parameters for crawling. Only the shoulder and the hip pitches are oscillating, and the amplitude of the shoulder is determined by the amplitude of the legs due to physical constraints (i.e. the horizontal distance covered by the hand during one step has to be equal to the one covered by the knee when the robot goes forward). The position of the shoulder roll and the elbow, and of the hip roll, is fixed during the stance but is modulated during swing according to Eq. 6.4 to avoid contact with the ground. The resulting trajectories can be seen on Fig. 6.6.

movements of the body during locomotion were neglected. To ensure that the distance between the knee and the hand touching the ground is constant, the amplitude for the arms were deduced from the leg amplitude according the kinematic model of the robot.

To make the robot turn, the torso roll angle is set to a non-zero value and the amplitude of the different limbs is modulated according to the new posture of the body. More precisely, the amplitudes of the inner pitch angles (i.e. the pitch angles of the arm and the leg on the side to which the robot is turning) will be smaller and the outer angles larger to compensate the fact that the outer side has to cover a larger distance than the inner one. The joint angle limits of the hips pitch constrain the amplitude range and the minimal radius of curvature that could be obtained (in simulation) was 1 m. Note that to improve steering in complex environments, backwards crawling could be implemented.

Fig 6.5 shows snapshots of the robot turning and the corresponding movie can be found at biorob2.epfl.ch/sd_movies, Movie 10.

6.4 Switching between crawling and reaching

When the robot is close enough to a target object to reach it, it stops crawling and goes to a rest position that is defined by the parameters g in Table 6.1, with all the oscillations “switched off” thanks to the Hopf bifurcation ($\mu_i < 0$). Once in this position the robot first lifts the arm that is going to reach for the object and then it reaches it, as illustrated by the snapshots on Fig. 6.7. The intermediate position (with the reaching arm lifted) has

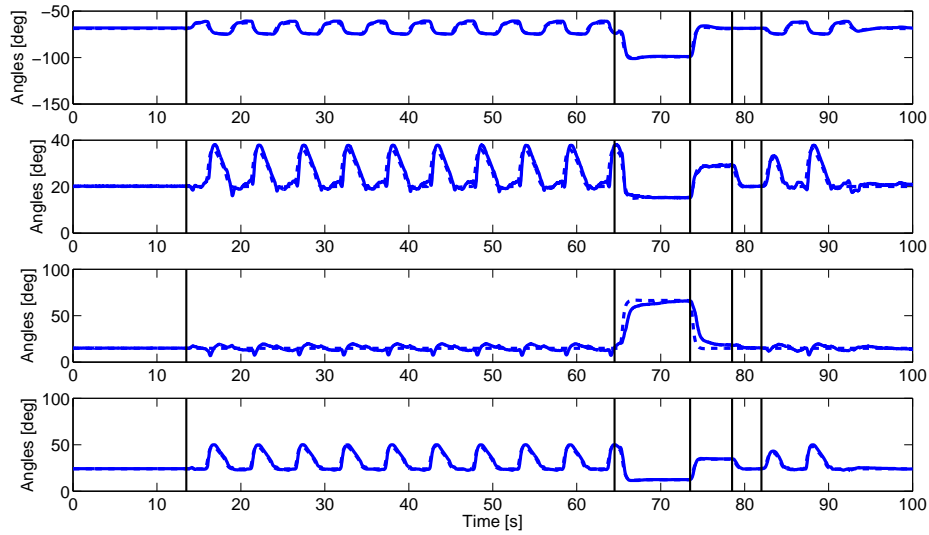


Figure 6.5: The iCub turning. Superimposed snapshots of the robot turning with a torso angle of 0.5 radians. Movie available at biorob2.epfl.ch/sd_movies, Movie 10.

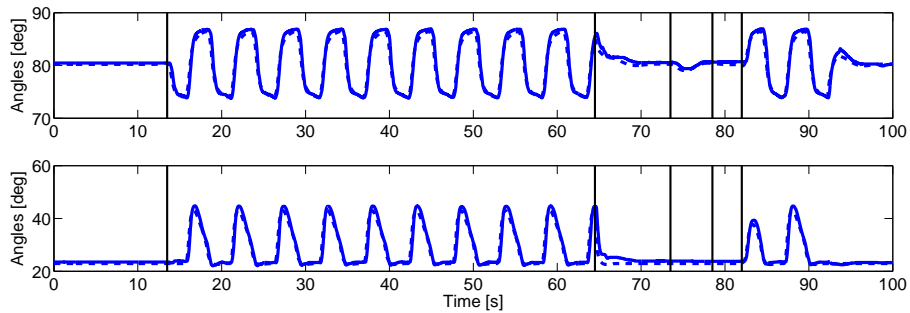
been added to avoid contact with the ground during the reaching movement. The object to be reached is indicated with a ARToolKitPlus marker and the target angle positions of the reaching arm are given by the inverse kinematics algorithm iKin mentioned above, while the other limbs stay in the same position. Note that as the mark is on the ground, the range of reachable positions for the arm is quite limited and thus the tolerated error for the inverse kinematics was set to a relatively high value (5cm).

Figs. 6.4 and 6.7 show snapshots of the robot crawling and reaching respectively. The corresponding movies are available at http://biorob2.epfl.ch/sd_movies, Movie 7 and Movie 8. Trajectories of the iCub crawling, going to the rest position, reaching for a mark and crawling again are depicted on Fig. 6.6. At $t \approx 13s$, the robot starts crawling. At $t \approx 64s$ the robot autonomously switch from the crawling behavior to the rest position because it is close enough to a mark to reach it. It goes to an intermediate position (in which the reaching arm – the right one here – is lifted above the ground) for 10s and then it reaches the mark on the ground (at $t \approx 74s$) until it is asked to crawl again ($t \approx 78s$): it first resumes to the rest position and start crawling again after a while (at $t \approx 82s$). Plain lines indicate the actual trajectories and dotted lines the desired ones, the tracking of the robot being quite good in general. *Top panel:* Trajectories of the four actively controlled DOFs of the right arm: shoulder pitch (blue), shoulder roll (green), shoulder yaw (red) and elbow (black). **a:** Trajectories of the four controlled DOFs of the right arm, from top panel to bottom one: the shoulder pitch, the shoulder roll, the shoulder yaw and the elbow. **b** Two DOFs of the left leg: the hip pitch (top panel) and the hip roll (bottom panel). Note that the hip yaw and the knee are kept in a constant position and are not shown here.

6.4. Switching between crawling and reaching



(a) Right Arm



(b) Left Leg

Figure 6.6: Trajectories of the (real) iCub crawling and reaching for the right arm and the left leg. Note that Figs. 6.4 and 6.7 show snapshots of the robot crawling and reaching respectively. Movies available at http://biorob2.epfl.ch/sd_movies, Movie 7 and Movie 8.

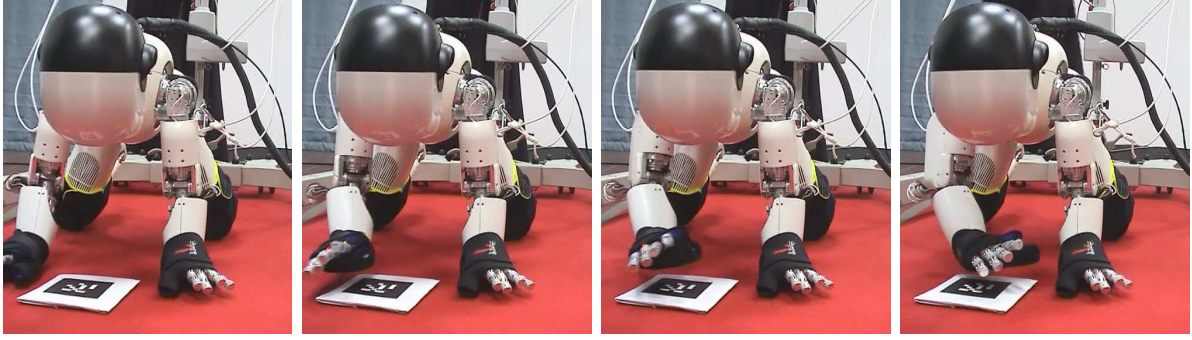


Figure 6.7: Snapshots of the robot reaching a mark. When the robot is close enough to a target object (here the marker on the ground), it stops crawling and goes back to the position defined by the discrete target (g in Tab. 6.1). Then it lifts the hand that is going to do the reaching movement (second snapshot). Finally it reaches for the target (third and fourth snapshots). Note that here the robot does not touch the ground as the maximal error tolerated was set to a high value (5cm). Typical reaching trajectories are depicted on Fig. 6.6. Movie available at http://biorob2.epfl.ch/sd_movies, Movie 8.

6.5 Contact feedback (Work by Ludovic Righetti)

Ludovic Righetti introduced a phase dependent sensory feedback in the UPG to increase locomotion stability on uneven terrains (Righetti and Ijspeert (2008)). In mammalian locomotion local sensory information such as load sensing on the extremities of a limb has an important role in the modulation of the onset of swing and stance phases (see Frigon and Rossignol (2006) for a review). He implemented in our crawling controller a similar feedback pathway. The dynamics of the oscillator and thus the policy generation is modified on line according to load sensing on the end effectors (hands and knees of the robot). As long as a limb supports the body weight the transition from stance to swing phase for this limb is delayed. A faster transition occurs in the case of an early leg unloading. In the case of swing to stance transition, a symmetric behavior is implemented. Transition is delayed as long as the limb does not touch the ground and is triggered in case of an early contact with the ground. We add to Eq 4.6 a feedback term to modulate the transitions that is defined as

$$\dot{z}_i = \dots + \begin{cases} -\text{sign}(z_i)F & \text{fast transitions} \\ -\omega(x_i - y_i)(+\text{couplings}) & \text{stop transition} \\ 0 & \text{otherwise} \end{cases} \quad (6.5)$$

where F ($= 10$ in our case) controls the speed of the transition. Fig. 6.8 shows the activation of the feedback depending on the phase of the limb and the resulting modification of the phase space of the oscillator. This feedback loop was tested in simulation in Righetti and Ijspeert (2008). Locomotion stability was improved on various terrains

6.5. Contact feedback (Work by Ludovic Righetti)

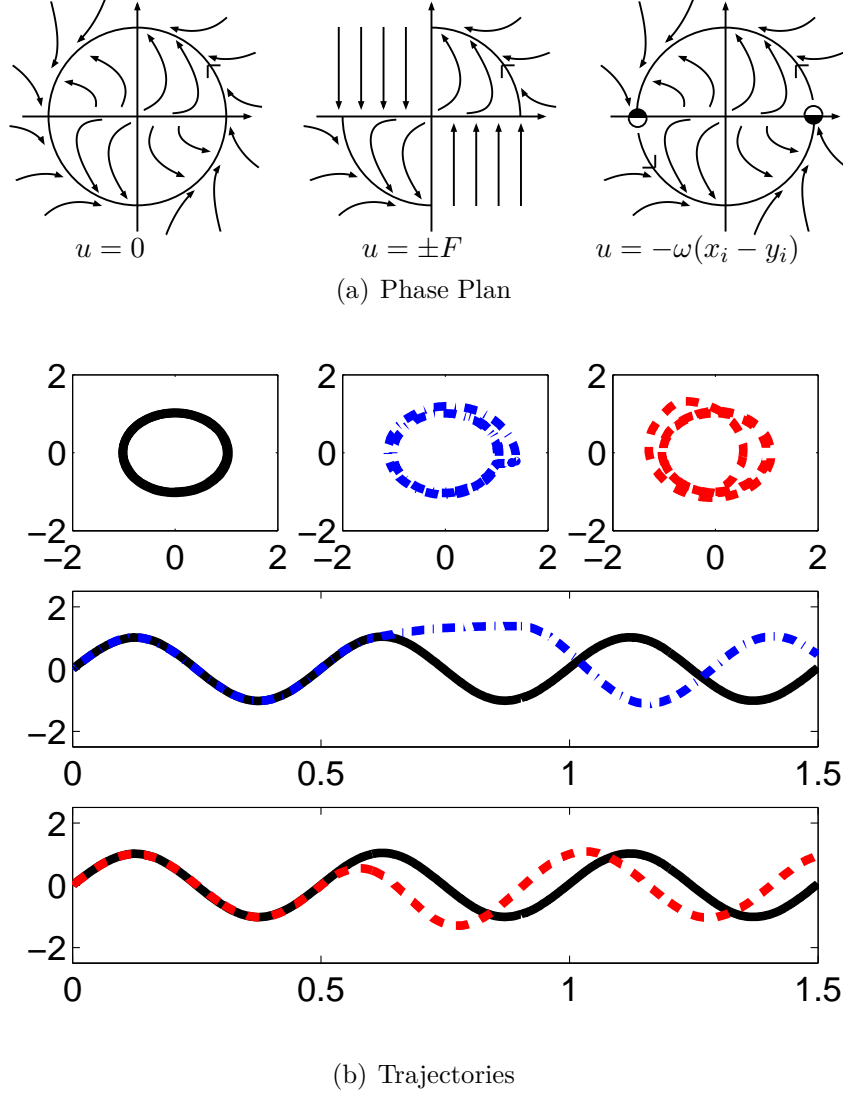


Figure 6.8: Feedback strategy for crawling. (a) Modulation of the phase plan by the feedback *Left panel*: normal phase plan of the the Hopf oscillator. *Middle panel*: Strategy for accelerating the transitions. The speed is increased by F ($=10$ in our case). *Right panel*: Strategy for slowing down transitions. The system is stopped by canceling the oscillatory terms. (b) Corresponding trajectories. *Top panel*: Trajectories in the phase plan corresponding the strategies in (a): no feedback (black), faster transition (red), slower transition (blue). *Middle panel*: Accelerated trajectory (blue, dash dotted line) compared to unperturbed trajectory (black, plain line) in time domain. *Bottom panel*: Accelerated trajectory (red, dash line) compared to unperturbed trajectory (black, plain line) in time domain.

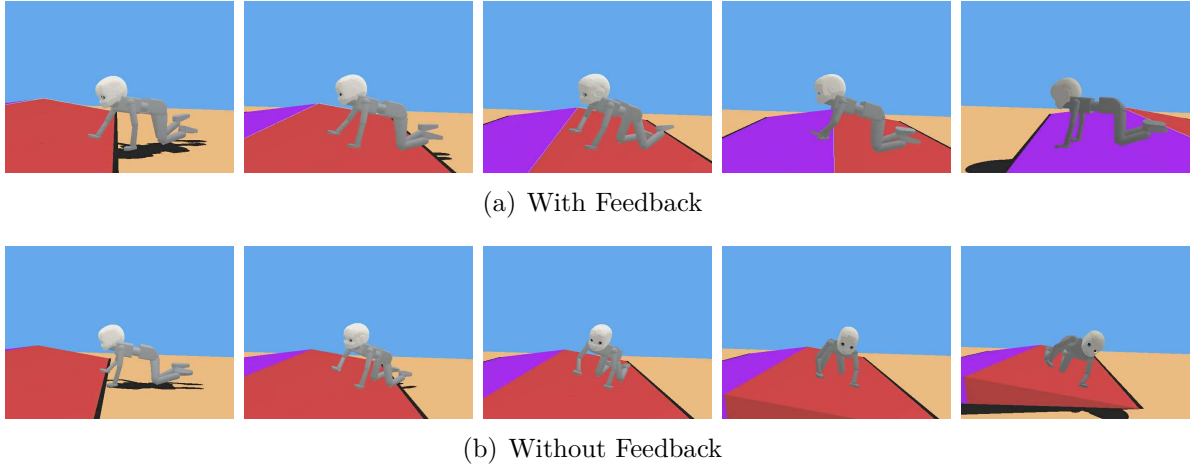


Figure 6.9: Clearing of a 10-degree slope with and without feedback. In both experiments, the robot was initially at the same position and snapshots were taken at the same time intervals. With the feedback, the length and duration of the steps are adapted to the terrain and it can be seen that robot successfully goes through the obstacle (top panel), while it fails to do it in open loop (bottom panel). Movie available at biorob2.epfl.ch/sd_movies, Movie 11.

such as slopes. Fig. 6.9 gives an illustration of the effect of feedback on the locomotion behavior.

This feedback uses only local information to change the control policies locally and as such is a first layer of adaptation in unpredicted environments. Such feedback pathways serve as an example to show the flexibility and versatility of the proposed architecture for on line trajectory generation. The detailed analysis of the feedback behavior is beyond the scope of this dissertation and can be found in Righetti and Ijspeert (2008); Righetti (2008).

6.6 Combination with a high-level path planning

A high level planner based on potential fields has been developed by Sebastien Gay (Gay et al (2010)) to illustrate how our controller can be used in a simple navigation task in a fully autonomous way. A representation map of the different positions of the obstacles and targets, acquired through a vision multi-object tracking module based on ARToolKitPlus, is turned into a potential field where obstacles and targets are represented by respectively positive and negative potentials (Khatib (1986)). Note that the standard implementation of the potential fields was slightly modified by Sebastien Gay to deal with multiple targets: the closer the robot is to a target, the more attractive it is. The trajectory is then given by the gradient of the surface.

6.7. Concluding remarks

The field of view of the robot was enhanced by coupling the head oscillators with the rest of the body in a way that the head and the eyes of the robot perform an oscillatory movement in phase with the crawling movement to scan the environment. The positions of the markers detected during one oscillation of the head and eyes are translated in the root reference frame of the robot using iKin (see Annex B) and used to construct a partial map of the direct surroundings of the robot. This partial map is the only information available to the robot to perform navigation. No external information (full map of the environment, self localization ...) is provided to the robot.

The command sent to the manager by the path planning module is to crawl with a certain desired angle of rotation. This angle correspond to the torso roll angle and is updated whenever it is required to follow the path. If the reaching module is active (i.e. if the module is launched), whenever the robot is close enough to target marker, the behavior will be switched to reaching and all the steering commands will be ignored during the completion of the reaching.

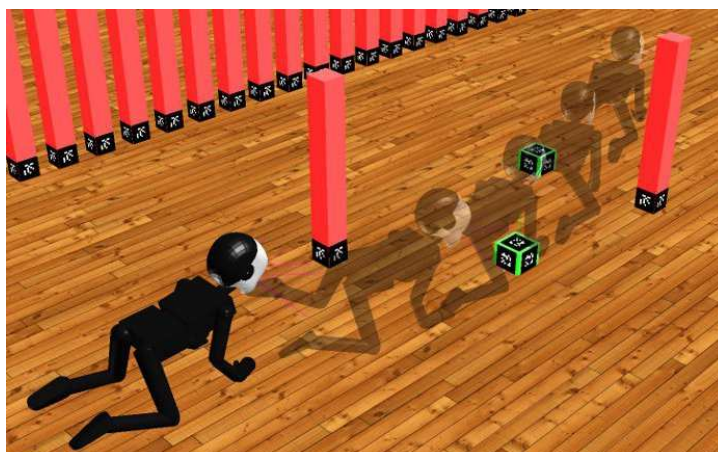


Figure 6.10: Snapshots of the iCub navigating in a complex environment. The robot is moving towards target object (small, green boxes), while avoiding obstacles (tall, red boxes). ARToolKit markers are used to track the object, with different markers for targets and obstacles. The information is sent to the path planning algorithm that computes a suitable path based on potential fields, this path is continuously updated. Movie available at http://biorob2.epfl.ch/sd_movies, Movie 12.

6.7 Concluding remarks

In this implementation, crawling is characterized by two control parameters: the duration of the stance – the amplitude of the oscillation being fixed in our case – and the torso roll angle for turning. The crawling behavior can thus be modulated in both ways: by

changing the duration of the stance to modulate the speed or by changing the torso roll angle to steer the robot. Thanks to the steering, the robot can evolve in a complex environment and avoid obstacles (as was illustrated with the high-level planner). In addition, the robot can easily switch between behaviors, such as crawling and reaching, depending on the visual sensory feedback. It can also adapt its current behavior, for instance to avoid obstacles and reach target objects, or by changing its step length to deal with tilted terrain. For now only steady-state crawling and reaching have been implemented on the real robot, but steering and changes of speed will be soon integrated. Note that a new implementation using gain scheduling is under development, in order to modulate on line the compliance of the robot according to the needs of the task (e.g. by having a lower stiffness during the swing than the stance to better handle the contact of the limbs end-effector with the ground).

CHAPTER 7

OUTLOOK

Humans are able to adapt their movements to almost any new situations in a very robust, seemingly effortless way. To explain both this adaptivity and robustness, a very promising perspective is the modular approach to movement generation: Movements results from combinations of a finite set of stable motor primitives organized at the spinal level. In this research we applied this concept of modularity to the control of robots with multiple degrees of freedom. We propose to decrease the complexity of the planning phase through the use of a combination of discrete and rhythmic motor primitives modeled by dynamical systems. We apply this control approach to different tasks, namely drumming, crawling and reaching.

The novelty here is that we address the generation of both discrete and rhythmic movements through the same system, a subject that as received little attention so far, as we reviewed in Chapter 3. Indeed, to the best of our knowledge, only two models for the simultaneous production of both discrete and rhythmic movements have been presented before, namely the models by De Rugy and Sternad (2003) and by Schaal et al (2000). The main differences between the system that we have designed and these models is that here our main focus is the application to robotic control rather than the reproduction of observations made in humans. Our focus was thus to design a simple model with few, explicit control parameters corresponding to main characteristics of the movement (that is, the discrete target, the frequency and the amplitude). Indeed, the model presented by De Rugy and Sternad (2003), later extended to bi-manual tasks by Ronsse et al (2009), is aimed at reproducing the key observations of the combination of discrete and rhythmic movements. It is based on a Matsuoka oscillator (Matsuoka (1985)) modeling the output of two coupled neurons, this output being transformed into a desired trajectory through the equation of the dynamics of the joint. While this model successfully reproduces key features of the human discrete and rhythmic movements, and of the interaction of the two movements, it is rather difficult to apply it to robotic control as the control parameters are not explicitly linked to key parameters of the end trajectory. Schaal et al (2000) proposed a rather complex system composed of two different motor primitives

with many parameters, that allows for the reproduction of signals recorded in the brain, the drawback being that all these parameters need to be tuned precisely. In addition, the dynamics of two primitives do not influence each other, which may lead to non natural output trajectories. Note that Schöner and Santos (2001) introduced a system where discrete movements are modeled as truncated rhythmic movements, which means that the whole trajectory is a stable solution (contrarily to most approaches where only the target is an attractor, see e.g. Bullock and Grossberg (1988) and Hersch et al (2008)). However, both types of movements can not be combined in a straightforward way.

One could argue that a major limitation of the approach is that the space of possible trajectories is limited by the chosen dynamics. It is not exactly true, as it is possible to constantly update the discrete target of the movement in order to obtain the desired movement, even though this would be merely a sort of by-pass of the motor primitives. In addition, the dynamics of the motor primitives can also be modulated according to the specific need of the task, as was done for instance for crawling by changing the expression of the frequency to have an independent control of the duration of the swing and the stance (Righetti and Ijspeert (2006a)). However, designing a dynamical system which solution is a predefined, desired trajectory is in general not an easy task. An interesting approach to this issue is the use of adaptive frequency oscillators (AFO) which are oscillators that can learn the frequencies of external signals through entrainment (Buchli and Ijspeert (2004), Righetti et al (2006)). It can be used for frequency analysis (Buchli et al (2008)) to decompose a signal, and Righetti and Ijspeert (2006b) used this approach to encode a complex trajectory into a sum of oscillators and successfully applied it to biped locomotion. Buchli and Ijspeert (2008) used this technique to develop an adaptive locomotion controller for compliant robots that can adapt to the body properties of the robot but also to different types of gaits. An interesting extension of the UPG presented in this dissertation would thus be to combine them with AFOs to learn periodic reference trajectories (using a constant offset). Further investigation would need to be done on how to integrate the discrete subsystem into the AFO framework. Another possible solution to generate more complex trajectories would be to use UPGs as a function basis for optimization given a reference trajectory. More precisely, one could search for a linear combination of UPGs that minimize the difference with the reference trajectory. The parameters to be optimized would then be the open parameters of the UPG as well as the number of UPGs needed [for a related approach with parabolic functions, see Polyakov et al (2009)].

As was illustrated in the application to crawling, the usage of a unique dynamical system for both discrete and rhythmic movements eases the switch between two totally different behaviors (crawling and reaching in our case): a unique term controlling the amplitude (μ in the equations) allows for transiting between discrete and rhythmic behaviors thanks to the Hopf bifurcation of the system. In our example, we used intermediate positions to ensure that the constraints induced by the close kinematics are fulfilled, but one could imagine to use more sophisticated techniques to compute the desired trajectory of

7.1. Contributions

the different limbs.

Finally, our approach to movement generation emphasizes adaptivity, in the sense that instead of using a purely predictive model, we extensively use feedback information to modulate the behavior of the robot according to its time-evolving environment (as for instance tilted terrain during crawling, moving obstacles during crawling, moving targets during reaching or moving drums while drumming). The feedback can be local and act directly on the CPGs (as for instance the contact feedbacks for both drumming and crawling) or require some high-level processing and have a behavioral effect (as the high-level planning in crawling). Such strategies created a tight coupling between the controller and the environment, making the whole architecture more robust to modeling imprecisions, perturbations or time-varying environment.

Defining quantitative benchmarks to compare our results with other approaches is difficult. Indeed, most of the applications to robotics based on dynamical system theory¹ are based on learning by demonstration (Ijspeert et al (2003), Gribovskaya and Billard (2008), Pastor et al (2009), Kober and Peters (2010)) and cannot be compared to the work presented here. To the best of our knowledge, it is the first time a system for generating both discrete and rhythmic movements (and based on motor primitives) is applied to real robots. In addition, infant crawling is qualitatively different from mammalian quadruped locomotion (notably because of the geometry of the limbs). Drumming has been studied before, but in different context, e.g., to study agent-object interaction (Williamson (1999)), learning from demonstration (Ijspeert et al (2002)) or human-robot interaction (Kose-Bagci et al (2010)).

Compared with traditional approaches based on optimization solely, the concept of motor primitives requires some a priori knowledge on the dynamics of the movements to design the UPG, but in turn it lightens the on line computational needs, making the architecture extremely well-suited for task needing fast control loops. The concept of motor primitives allows for a trade-off between the constraints you want to impose to your trajectories and the time of calculation required for the trajectories.

7.1 Contributions

Here we recall the different contributions that were stated in the introduction and discuss them a bit further.

- **An extensive review on the generation of discrete and rhythmic movements in vertebrates in the perspective of motor primitives**

Discrete and rhythmic movements are often considered separately in theories on motor control. The few studies that exist on their combination either consider

¹If we exclude the work targeting at explaining biological processes, in which case the performance criterion is plausibility rather than efficiency.

them at the motor cortex level or at the muscle level. Yet, an increasing number of studies have brought to light the existence of spinally encoded motor primitives of movements. We thus review the existing literature on this concept to try to gain insights on the generation of discrete and rhythmic movements at the spinal level.

- **A classification of four possible types of model for the generation of discrete and rhythmic movements**

Thanks to the concept of motor primitives, we can easily decouple the planning of the movement (i.e. the choice of its characteristics) from the actual generation of the desired trajectories. This allows us to define four different types of models for the generation of discrete and rhythmic movements. The types of movements are illustrated by mathematical models existing in the literature.

- **A control architecture based on motor primitives that allows for**
 - **the generation of both discrete and rhythmic movements**
 - **the switch between these two types of movements**
 - **the integration of feedback modulations**

The model that we have presented can be seen as a simple trajectory generator for both discrete and rhythmic movements that is easy to control and that can be modulated on line according to new control commands and/or feedback. Such a generator drastically reduces the planning as only the key characteristics of the movements need to be specified, namely the target of the discrete movements g_i , and the amplitude $\sqrt{[\mu_i]^+}$ and the frequency ω_i of the rhythmic one. In addition, the global attractiveness of the solutions ensures robustness against perturbations, but also the capacity of the system to adapt to changing environments through feedback information. The three main advantages of the approach are that (i) the planning phase is simplified thanks to the motor primitives, in the sense that the control commands that are required are reduced to the key characteristics of the movement (the target for discrete movements and the amplitude and frequency for rhythmic movements), (ii) switching between behaviors is made easier by the fact that the same system can be used for all kind of tasks, either discrete and rhythmic, and (iii) the dynamics of the motor primitives can be modulated by sensory feedback in order to obtain an adaptive behaviors. In addition, this method has a low computational cost and is well-fitted for applications requiring fast control loops.

- **The implementation of interactive, closed-loop drumming on the iCub robot, as well as a previous implementation on the HOAP-2 robot**

Drumming was implemented on the HOAP-2 and the iCub robot. In the first implementation, the task was defined by musical scores expressed as matrices. The architecture was suitable for fast control loops (1 ms), and in simulation, the robot

7.1. Contributions

could be coupled to its environment through a feedback loop on the position of the drums. In the second application, on the iCub, the score can be defined online by any user through a graphical user interface, showing the robustness of the system against parameter changes. The feedback controller on the position of the drums was improved by adding vision, making it suitable to be implemented on the real robot. In addition a second type of feedback was introduced to deal with the contact with the drums. This time the feedback directly acts on the CPGs to modulate the trajectories. It could be seen that thanks to the attractive properties of the system, it was able to resume to the normal trajectories after the perturbation (i.e. the feedback term) vanishes.

- **The implementation of autonomous, closed-loop crawling and reaching on the iCub robot**

The implementation of crawling nicely illustrates the fact that the architecture allows for the generation of a wide range of behaviors from crawling to reaching, but also that switching between these two behaviors is extremely simple. The fact that the parameters of the primitives are explicit allowed to couple the system with a high level planner in a straightforward way. The ability to modulate the trajectories of the CPG was illustrated by the addition of a simple term to control independently the swing and the stance, but also through the addition of the feedback controller based on contact information.

Future work

As it was said in the introduction, we think that a major improvement of the current approach would be to combine it with modern control techniques to deal with multiple constraints (e.g. balance, closed loop chains, obstacles, ...), such as the whole-body control approach proposed by Sentis and Khatib (2005) for instance.

Combining the architecture with optimal control techniques, such as the work by Todorov et al (2005) or Polyakov et al (2009) that were presented in the state of the art, could also bring interesting results. Indeed one could imagine that instead of computing trajectories with desired dynamics every time, the dynamics could be embedded in a set of motor primitives that would provide a set of basis functions for the space of trajectories. The only task of the optimizer would be then to select the appropriate parameters for the primitives. In addition, the fact that the trajectories are robust against perturbation could be used to directly modulate the dynamics according the environmental constraints rather than computing an entirely new trajectory.

Finally, in terms of implementation, several improvements could be done. For the drumming implementation, we will implement the feedback on the position of the drums on the real robot. We are also working on an implementation of the crawling based

on impedance control. We would like to test the steering and the change of speed of crawling in the next future. We also would like to develop a system for balance control during locomotion and reaching that would simply use the discrete system to correct the trajectories according to the needs of the task. In addition techniques such as the one presented in Ijspeert et al (2003) and Hersch et al (2008) could be used to deal with joint limits and singular configurations.

APPENDIX A

DERIVATION OF THE RADIUS OF EQ. 4.8, CHAPTER 4

We want to find the analytical solutions of the Eq. A.2 in the following system:

$$\dot{m} = C(\mu - m) \quad (\text{A.1})$$

$$\dot{r} = \frac{A}{|\mu|} (m - r^2) r \quad (\text{A.2})$$

$$\dot{\theta} = \omega \quad (\text{A.3})$$

with $r \in \mathcal{R}^+$ and $\theta \in \mathcal{R}$.

We solve Eq.A.2 for the case where $\mu > 0$ (limit cycle system). It can be solved as a Bernoulli equation, indeed we have an equation of the form:

$$\frac{\dot{r}}{r^3} - \frac{A\mu}{|\mu|r^2} = -\frac{A}{|\mu|}$$

Using the substitution $v = r^{-2}$ (we assume $r(t) \neq 0, \forall t$, we obtain

$$\dot{v} + 2\frac{A}{|\mu|}mv = 2\frac{A}{|\mu|}. \quad (\text{A.4})$$

We first solve the homogeneous equation, i.e

$$\dot{v} + 2\frac{A}{|\mu|}mv = 0, \quad (\text{A.5})$$

to obtain:

$$v(t) = C_v e^{-2\frac{A}{|\mu|}(\mu t + (\mu - m_0)e^{-t})}. \quad (\text{A.6})$$

A particular solution is given by $v(t) = \mu^{-1}$ if we assume $m(t) = \mu$. The general solution can thus be written as

$$r^2 = \frac{\mu}{1 + \mu C_r e^{-\frac{2A}{|\mu|}(\mu t + (\mu - m_0)e^{-t})}} \quad (\text{A.7})$$

APPENDIX B

INFORMATION ON THE ROBOTS

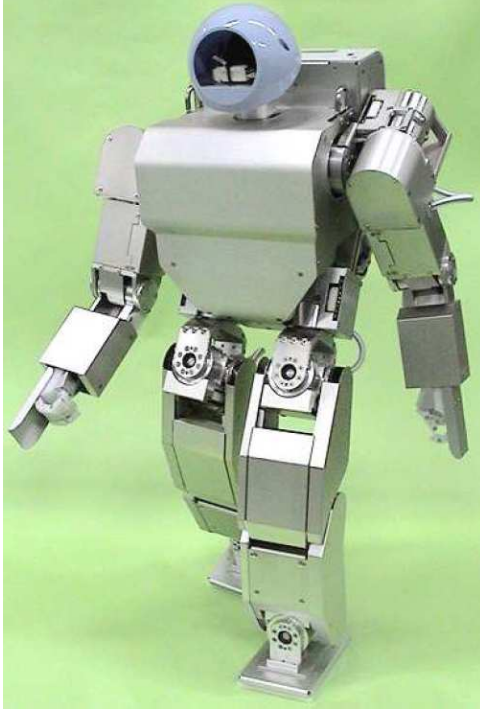
We briefly summarize here some information that might help to better understand the implementation of drumming (Chapter 5) and crawling (Chapter 6). We start by a description of the two robot we used, the HOAP-2 and the iCub, in terms of hardware and software and then we present the Webots simulator.

B.1 HOAP-2

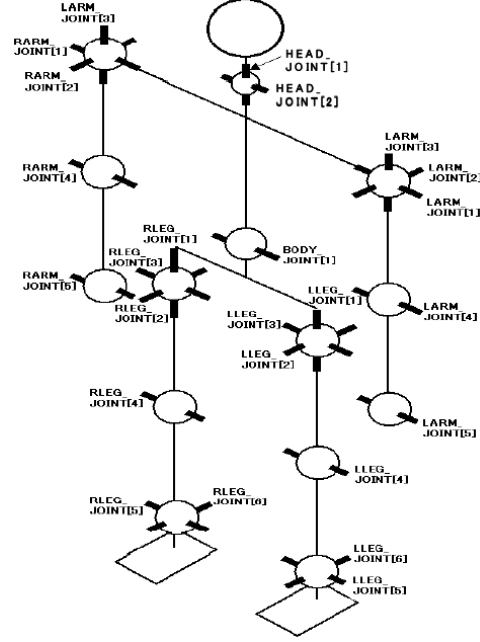
The HOAP-2 is the second version of a commercial humanoid robot developed by Fujitsu. It was released in 2003 and a third version, the HOAP-3 followed in 2005. The software language is C/C++ and it runs on RT-Linux computer, with a control loop of 1 ms. The robot is 50 cm height and weight 7 kg. It has 25 degrees of freedom, its general structure is depicted on Fig. B.1.

B.2 iCub

The iCub robot has been developed in the framework of the RobotCub project, a 5 year-long EU-funded project that ended in January 2010. Its goals were twofold: first, to develop a humanoid robot – the iCub – of the size of a 3.5 years old infant, and second, to use this platform to study cognition and its development (see Tsagarakis et al (2007) for instance). All the software developed during this project for the iCub robot, and notably the code for crawling and drumming that will be presented below, is open source. The software is based on the open source library YARP developed by Fitzpatrick et al (2008) to support software development and integration in robotics.



(a) The HOAP-2 robot



(b) Structure of the HOAP-2

Figure B.1: The HOAP-2 robot and its structure.

B.2.1 Hardware

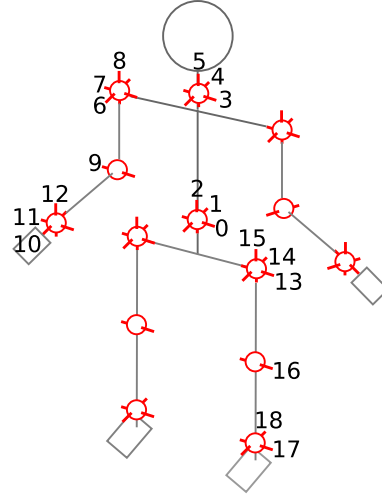
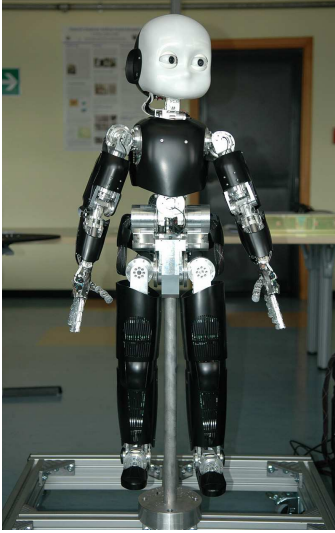
The robot has 53 degrees of freedom (DOFs): 6 for each leg, 16 for each arm (among which 9 for each hand), 3 for the torso and 6 for the head. Most of the DOFs axes and their names are depicted on Figure B.2.

B.2.2 iCub Software

The iCub software is built collaboratively by the different actors of the project, it is open source and can be found on the project website (www.robotcub.org). Integration and compatibility issues are solved via the use of a common middleware, YARP (Metta et al (2006)), and the use of a common version controlled repository for the whole project. In this paragraph we present this iCub middleware as well as the main library that we use for kinematics computation, iKin.

The iCub software architecture is based on YARP, an inter-process communication layer, which enables complete abstraction of the communication protocol between different software modules. Each module streams its output data through YARP ports, and these building blocks can be interconnected regardless of their physical location on

B.3. Webots



- 0 torso pitch
- 1 torso roll
- 2 torso yaw
- 3 neck pitch
- 4 neck roll
- 5 neck yaw
- 6 shoulder pitch
- 7 shoulder roll
- 8 shoulder yaw
- 9 elbow
- 10 wrist prosup
- 11 wrist pitch
- 12 wrist yaw
- 13 hip pitch
- 14 hip roll
- 15 hip yaw
- 16 knee
- 17 ankle pitch
- 18 ankle roll

Figure B.2: the iCub and its structure. Schematic of the dofs of the iCub (excluding the dofs of the hands and eyes).

the network (same computer, Ethernet network, etc.). iCub capabilities are thus implemented as a set of modules that can be easily connected together through YARP ports.

The software for the iCub comes with a set of kinematics libraries called iKin, developed by U. Pattacini. It allows forward and inverse kinematics computations on any subchain of the iCub degrees of freedom. The forward kinematics library uses standard Denavit-Hartenberg convention to enable the projection of a position to the reference frame of any part of the robot. It can be used for instance to project the position of an object in the camera reference frame to the root reference frame of the robot, or to check for internal collisions. The inverse kinematics library uses the IPOPT library (Wächter and Biegler (2006)) to solve the non-linear inverse kinematics problem with N DOFs under the set of constraints defined by the limits of each joint. A maximum error as well as a maximum number of iteration of the optimization algorithm can be set for a compromise between precision and computational complexity.

B.3 Webots

WebotsTM (Michel (2004)) is the ODE-based simulator that we use to develop the software code for both drumming and crawling. A model of the iCub was developed with

Appendix B. Information on the robots

respect the Denavit-Hartenberg parameters of the real iCub as well as the joints limits and maximum torque of the motors. A YARP interface similar to the iCub robot interface was also developed, so that the same YARP modules can be used on the simulator and on the real robot without any modification. This software is freely available on the RobotCub website.

APPENDIX C

LIST OF CONTRIBUTORS

Here is a list of the persons who have contributed to the implementation of the applications:

1. Ludovic Righetti
 - (a) Development of the first version of the controller for crawling (CPG design, independent control of swing and stance)
 - (b) Development of the contact feedback policy for crawling
 - (c) Participation in the software development and implementation for both drumming and crawling with the iCub
2. Sebastien Gay
 - (a) Software development of the visual tracking system for both drumming and crawling
 - (b) Adaptation of the iKin library for both drumming and crawling
 - (c) Help with the implementation of drumming and crawling with the iCub
3. Francesco Nori
 - (a) Support for the iCub platform
 - (b) Help with the implementation of drumming and crawling with the iCub
4. Lorenzo Natale
 - (a) Support for the YARP library
 - (b) Help with the implementation of drumming with the iCub
5. Cristina Santos
 - (a) Collaboration for the implementation of drumming on the HOAP-2

BIBLIOGRAPHY

- Adamovich S, Levin M, Feldman A (1994) Merging different motor patterns: Coordination between rhythmical and discrete single-joint movements. *Exp Brain Research* 99(2):325–337
- Aoi S, Tsuchiya K (2007) Adaptive behavior in turning of an oscillator-driven biped robot. *Auton Robots* 23(1):37–57
- Ashe J (2005) What is coded in the primary cortex? In: Riehle A, Vaadia E (eds) *Motor Cortex In Voluntary Movements*, CRC Press
- Barbeau H, Rossignol S (1994) Enhancement of locomotor recovery following spinal cord injury. *Curr Opin Neurol* 7(6):517–24
- Bernstein N (1967) *The Co-ordination and Regulation of Movements*. Oxford, UK: Pergamo
- Bizzi E, Accornero N, Chapple W, Hogan N (1984) Posture control and trajectory formation during arm movement. *J Neurosci* 4(11):2738–2744
- Bizzi E, Mussa-Ivaldi FA, Giszter S (1991) Computations underlying the execution of movement: a biological perspective. *Science* 253(5017):287–91
- Bizzi E, Cheung VCK, d’Avella A, Saltiel P, Tresch M (2008) Combining modules for movement. *Brain Res Rev* 57(1):125–33
- Bridgeman B (2007) Efference copy and its limitations. *Computers in Biology and Medicine* 37(7):924–929
- Brown T (1912) The factors in rhythmic activity of the nervous system. *Proceedings of the Royal Society of London Series* 85(579):278–289
- Buchli J, Ijspeert A (2004) Distributed central pattern generator model for robotics application based on phase sensitivity analysis. In: Ijspeert A, Murata M, Wakamiya N (eds) *Biologically Inspired Approaches to Advanced Information Technology: First International Workshop, BioADIT 2004*, Springer Verlag Berlin Heidelberg, Lecture Notes in Computer Science, vol 3141, pp 333–349

- Buchli J, Ijspeert AJ (2008) Self-organized adaptive legged locomotion in a compliant quadruped robot. *Autonomous Robots* 25(4):331–347
- Buchli J, Righetti L, Ijspeert A (2008) Frequency analysis with coupled nonlinear oscillators. *Physica D: Nonlinear Phenomena* 237:1705–1718
- Bullock D, Grossberg S (1988) The VITE model: a neural command circuit for generating arm and articulator trajectories. In: Kelso J, Mandell A, Shlesinger M (eds) *Dynamic patterns in complex systems*, Singapore: World Scientific, pp 206–305
- Bullock D, Grossberg S (1989) Volitional Action, Amsterdam:North-Holland, chap VITE and FLETE: Neural Models for trajectory formation and postural control, pp 253–297
- Capaday C (2002) The special nature of human walking and its neural control. *TRENDS in Neurosciences* 25(7):370–376
- Cheng J, Stein R, Jovanovic K, Yoshida K, Bennett D, Han Y (1998) Identification, localization, and modulation of neural networks for walking in the mudpuppy (necturus maculatus) spinal cord. *The Journal of Neuroscience* 18(11):4295–4304
- Cohen A, Wallen P (1980) The neural correlate of locomotion in fish. “Fictive swimming” induced in a in vitro preparation of the lamprey spinal cord. *Exp Brain Res* 41:11–18
- Crespi A, Lachat D, Pasquier A, Ijspeert AJ (2008) Controlling swimming and crawling in a fish robot using a central pattern generator. *Autonomous Robots* 25(1-2):3–13
- d’Avella A, Portone A, Fernandez L, Lacquaniti F (2006) Control of Fast-Reaching movements by muscle synergy combinations. *J Neurosci* 26(30):7791–7810
- De Rugy A, Sternad D (2003) Interaction between discrete and rhythmic movements: reaction time and phase of discrete movement initiation during oscillatory movements. *Brain Research* 994(2):160–174
- Degallier S, Ijspeert A (2010) Modeling discrete and rhythmic movements through motor primitives: A review. *Accepted for publication in Biological Cybernetics*
- Degallier S, Santos CP, Righetti L, Ijspeert A (2006) Movement generation using dynamical systems: a humanoid robot performing a drumming task. In: *IEEE-RAS Inter. Conf. on Humanoid Robots*, pp 512–517
- Degallier S, Righetti L, Ijspeert A (2007) Hand placement during quadruped locomotion in a humanoid robot: A dynamical system approach. In: *IEEE-RAS International Conference on Intelligent Robots and Systems (IROS07)*

Bibliography

- Degallier S, Righetti L, Natale L, Nori F, Metta G, Ijspeert A (2008) A modular bio-inspired architecture for movement generation for the infant-like robot icub. In: Proceedings of the second IEEE RAS / EMBS International Conference on Biomedical Robotics and Biomechatronics, BioRob
- Degallier S, Righetti L, Gay S, Ijspeert AJ (2010) Towards simple control for complex, autonomous robotic applications: Combining discrete and rhythmic motor primitives. *Submitted to Autonomous Robots*
- Delcomyn F (1980) Neural basis of rhythmic behavior in animals. *Science* 210:492–498
- Delvolvé I, Branchereau P, Dubuc R, Cabelguen JM (1999) Fictive rhythmic motor patterns induced by NMDA in an in vitro brain stem-spinal cord preparation from an adult urodele. *Journal of Neurophysiology* 82:1074–1077
- Dietz V, Harkema SJ (2004) Locomotor activity in spinal cord-injured persons. *J Appl Physiol* 96(5):1954–1960
- Dietz V, Muller R, Colombo G (2002) Locomotor activity in spinal man: significance of afferent input from joint and load receptors. *Brain* 125(12):2626–2634
- Dimitrijevic MR, Gerasimenko Y, Pinter MM (1998) Evidence for a spinal central pattern generator in humans. *Annals of the New York Academy of Sciences* 860:360–376
- Edgerton VR, Tillakaratne NJ, Bigbee AJ, de Leon RD, Roy RR (2004) Plasticity of the spinal neural circuitry after injury. *Annual Review of Neuroscience* 27(1):145–167
- Elble R, Higgins C, Hughes L (1994) Essential tremor entrains rapid voluntary movements. *Exp Neurol* 126:138–143
- Feldman A (2009) New insights into action perception coupling. *Experimental Brain Research* 194(1):39–58
- Fitzpatrick P, Metta G, Natale L (2008) Towards long-lived robot genes. *Robot Auton Syst* 56(1):29–45
- Forssberg H (1985) Ontogeny of human locomotor control i. infant stepping, supported locomotion and transition to independent locomotion. *Experimental Brain Research* 57(3):480–493
- Frigon S, Rossignol S (2006) Experiments and models of sensorimotor interactions during locomotion. *Biological Cybernetics* 95(6):607–627
- Gandevia S, Burke D (1992) Does the nervous system depend on kinesthetic information to control natural limb movements? *Behav Brain Sci* 15:614–632

- Gaudiano P, Grossberg S (1992) Adaptive vector integration to endpoint: Self-organizing neural circuits for control of planned movement trajectories. *Human Movement Science* 11(1-2):141–155
- Gay S, Degallier S, Pattacini U, Ijspeert A, Santos J (2010) Integration of vision and central pattern generator based locomotion for path planning of a nonholonomic crawling humanoid robot. In: *Proceedings of the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2010)*, Tapei
- Georgopoulos AP (1996) On the translation of directional motor cortical commands to activation of muscles via spinal interneuronal systems. *Brain Res Cogn Brain Res* 3(2):151–5
- Gibet S, Kamp JF, Poirier F (2004) Gesture analysis: Invariant laws in movement. In: *Gesture-Based Communication in Human-Computer Interaction*, Springer Berlin / Heidelberg, vol 2915/2004, pp 1–9
- Giszter SF, Mussa-Ivaldi FA, Bizzi E (1993) Convergent force fields organized in the frog’s spinal cord. *J Neurosci* 13(2):467–91
- Goodman D, Kelso J (1983) Exploring the functional significance of physiological tremor: A biospectroscopic approach. *Experimental Brain Research* 49:419–431
- Graziano MSA, Taylor CSR, Moore T, Cooke DF (2002) The cortical control of movement revisited. *Neuron* 36:349–362
- Gribovskaya E, Billard A (2008) Combining Dynamical Systems Control and Programming by Demonstration for Teaching Discrete Bimanual Coordination Tasks to a Humanoid Robot. In: *Proceedings of 3rd ACM/IEEE International Conference on Human-Robot Interaction, HRI’08*, Amsterdam, March 12-15 2008
- Grillner S (1985) Neurobiological bases of rhythmic motor acts in vertebrates. *Science* 228(4696):143–149
- Grillner S (2006) Biological pattern generation: The cellular and computational logic of networks in motion. *Neuron* 52(5):751–766
- Grillner S, Zangger P (1984) The effect of dorsal root transection on the efferent motor pattern in the cat’s hindlimb during locomotion. *Acta Physiologica Scandinavica* 120(3):393–405
- Guiard Y (1993) On Fitts’s and Hooke’s laws: Simple harmonic movement in upper-limb cyclical aiming. *Acta Psychol (Amst)* 82:139–159

Bibliography

- Haiss F, Schwarz C (2005) Spatial segregation of different modes of movement control in the whisker representation of rat primary motor cortex. *J Neurosci* 25(6):1579–87
- Hanna JP, Frank JI (1995) Automatic stepping in the pontomedullary stage of central herniation. *Neurology* 45(5):985–986
- Hersch M, Guenter F, Calinon S, Billard A (2008) Dynamical System Modulation for Robot Learning via Kinesthetic Demonstrations. *IEEE Transactions on Robotics* 24(6):1463–1467
- Hogan N, Sternad D (2007) On rhythmic and discrete movements: reflections, definitions and implications for motor control. *Experimental Brain Research* 181(1):13–30
- Ijspeert A, Nakanishi J, Schaal S (2002) Learning rhythmic movements by demonstration using nonlinear oscillators. In: *Proceedings of the IEEE/RSJ Int. Conference on Intelligent Robots and Systems (IROS2002)*, pp 958–963
- Ijspeert AJ (2008) Central pattern generators for locomotion control in animals and robots: a review. *Neural Networks* 21(4):642–653
- Ijspeert AJ, Nakanishi J, Schaal S (2003) Learning attractor landscapes for learning motor primitives. In: S Becker ST, Obermayer K (eds) *Neural Information Processing Systems 15 (NIPS2002)*, pp 1547–1554
- Ijspeert AJ, Crespi A, Ryczko D, Cabelguen JM (2007) From swimming to walking with a salamander robot driven by a spinal cord model. *Science* 315(5817):1416–1420
- Jeannerod M (1988) *The neural and the behavioural organization of goal directed movements*. Oxford Science, Oxford
- Kalakrishnan M, Buchli J, Pastor P, Mistry M, Schaal S (2010) fast, robust quadruped locomotion over challenging terrain. In: *IEEE International Conference on Robotics and Automation*
- Kandel ER, Schwartz J, Jessell TM (2000) *Principles of Neural Science*. Mc Graw Hill
- Kargo W, Giszter S (2000) Rapid correction of aimed movements by summation of force-field primitives. *J Neurosci* 20(1):409–426
- Kawato M (1996) Learning internal models of the motor apparatus. In: JR Bloedel TESW (ed) *The Acquisition of Motor Behavior in Vertebrates*, Cambridge MA: MIT Press, pp 409–430
- Khatib O (1986) Real-time obstacle avoidance for manipulators and mobile robots. *Int J Rob Res* 5(1):90–98

- Kimura H, Fukuoka Y, Cohen AH (2007) Adaptive dynamic walking of a quadruped robot on natural ground based on biological concepts. *The International Journal of Robotics Research* 26(5):475–490
- Kober J, Peters J (2010) Imitation and reinforcement learning. *Robotics and Automation Magazine, IEEE* 17(2):55–62
- Kose-Bagci H, Dautenhahn K, Syrdal DS, Nehaniv CL (2010) Drum-mate: interaction dynamics and gestures in human-humanoid drumming experiments. *Connect Sci* 22(2):103–134
- Krouchev N, Kalaska JF, Drew T (2006) Sequential activation of muscle synergies during locomotion in the intact cat as revealed by cluster analysis and direct decomposition. *J Neurophysiol* 96(4):1991–2010
- Liu D, Todorov E (2009) Hierarchical optimal control of a 7-dof arm model. In: *proceedings of the 2nd IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning*, pp 50–57
- Marder E, Bucher D (2001) Central pattern generators and the control of rhythmic movements. *Curr Biol* 11(23):R986–96
- Matsuoka K (1985) Sustained oscillations generated by mutually inhibiting neurons with adaptation. *Biol Cybern* 52:367–376
- Maufroy C, Kimura H, Takase K (2008) Towards a general neural controller for quadrupedal locomotion. *Neural Networks* 21(4):667–681
- Metta G, Fitzpatrick P, Natale L (2006) Yarp: Yet another robot platform. *International Journal of Advanced Robotics Systems*, special issue on Software Development and Integration in Robotics 3(1)
- Miall RC, Ivry R (2004) Moving to a different beat. *Nat Neurosci* 7(10):1025–6
- Michaels C, Bongers R (1994) The dependance of discrete movements on rhythmic movements: simple rt during oscillatory tracking. *Hum Mov Sci* 13:473–493
- Michel O (2004) Webots TM: Professional mobile robot simulation. *International Journal of Advanced Robotic System* 1:39–42
- Morasso P (1981) Spatial control of arm movements. *Experimental Brain Research* 42(2):223–227
- Morimoto J, Endo G, Nakanishi J, Cheng G (2008) A biologically inspired biped locomotion strategy for humanoid robots: modulation of sinusoidal patterns by a coupled oscillator model. *IEEE Transactions on Robotics*

Bibliography

- Morishita I, Yajima A (1972) Analysis and simulation of networks of mutually inhibiting neurons. *Biological Cybernetics* 11(3):154–165
- Mussa-Ivaldi FA (1999) Modular features of motor control and learning. *Current Opinion in Neurobiology* 9(6):713–717
- Mussa-Ivaldi FA, Bizzi E (2000) Motor learning through the combination of primitives. *Philos Trans R Soc Lond B Biol Sci* 355(1404):1755–69
- Mussa-Ivaldi FA, Giszter SF, Bizzi E (1994) Linear combinations of primitives in vertebrate motor control. *Proc Natl Ac Sci USA* 91:7534–7538
- Overduin SA, d’Avella A, Roh J, Bizzi E (2008) Modulation of muscle synergy recruitment in primate grasping. *J Neurosci* 28(4):880–892
- Pastor P, Hoffmann H, Asfour T, Schaal S (2009) Learning and generalization of motor skills by learning from demonstration. In: international conference on robotics and automation (icra2009)
- Pearson KG (2000) Neural adaptation in the generation of rhythmic behavior. *Ann Rev Physiol* 62:723–753
- Peiper A, Nagler B (1963) *Cerebral Function in Infancy and Childhood*. Pitman Medical Publ, London
- Polyakov F, Stark E, Drori R, Abeles M, Flash T (2009) Parabolic movement primitives and cortical states: merging optimality with geometric invariance. *Biological Cybernetics* 100(2):159–184
- Popovic MB, Goswami A, Herr H (2005) Ground reference points in legged locomotion: Definitions, biological trajectories and control implications. *The International Journal of Robotics Research* 24(12):1013 –1032
- Pratt J, Chew CM, Torres A, Dilworth P, Pratt G (2001) Virtual model control: An intuitive approach for bipedal locomotion. *The International Journal of Robotics Research* 20(2):129–143
- Reiss RF (1962) A theory and simulation of rhythmic behavior due to reciprocal inhibition in small nerve nets. In: *Proceedings of the May 1-3, 1962, spring joint computer conference, ACM, San Francisco, California*, pp 171–194
- Righetti L (2008) Control of legged locomotion using dynamical systems. PhD thesis, EPFL, Lausanne

- Righetti L, Ijspeert A (2006a) Design methodologies for central pattern generators: an application to crawling humanoids. In: Proceedings of Robotics: Science and Systems, Philadelphia, USA
- Righetti L, Ijspeert A (2006b) Programmable central pattern generators: an application to biped locomotion control. In: Proceedings of the 2006 IEEE International Conference on Robotics and Automation
- Righetti L, Ijspeert A (2008) Pattern generators with sensory feedback for the control of quadruped locomotion. In: Proceedings of the 2008 IEEE International Conference on Robotics and Automation (ICRA 2008), pp 819–824
- Righetti L, Buchli J, Ijspeert A (2006) Dynamic hebbian learning in adaptive frequency oscillators. *Physica D* 216(2):269–281
- Ronsse R, Sternad D, Lefèvre P (2009) A computational model for rhythmic and discrete movements in uni- and bimanual coordination. *Neural Computation* 21(5):1335–1370
- Ronsse R, Vitiello N, Lenzi T, van den Kieboom J, Carrozza MC, Ijspeert AJ (2010) Human-robot synchrony: flexible assistance using adaptive oscillators. *Submitted to IEEE trans in neural systems and rehab engineering*
- Rossignol S, Schwab M, Schwartz M, Fehlings MG (2007) Spinal cord injury: Time to move? *J Neurosci* 27(44):11,782–11,792
- Saltiel P, Tresch MC, Bizzi E (1998) Spinal cord modular organization and rhythm generation: an nmda iontophoretic study in the frog. *J Neurophysiol* 80(5):2323–39
- Saltiel P, Wyler-Duda K, d’Avella A, Ajemian RJ, Bizzi E (2005) Localization and connectivity in spinal interneuronal networks: the adduction-caudal extension-flexion rhythm in the frog. *J Neurophysiol* 94(3):2120–38
- Schaal S, Kotosaka S, Sternad D (2000) Nonlinear dynamical systems as movement primitives. In: International Conference on Humanoid Robotics (Humanoids00), Springer, pp 117–124
- Schaal S, Sternad D, Osu R, Kawato M (2004) Rhythmic arm movement is not discrete. *Nat Neuroscience* 7(10):1136–1143
- Schöner G, Santos C (2001) Control of movement time and sequential action through attractor dynamics: A simulation study demonstrating object interception and coordination. In: *Neurons, Networks, and Motor Behavior*
- Sentis L, Khatib O (2005) Synthesis of whole-body behaviors through hierarchical control of behavioral primitives. *International Journal of Humanoid Robotics* 2(4):505–518

Bibliography

- Shadmehr R, Mussa-Ivaldi FA (1994) Adaptive representation of dynamics during learning of a motor task. *J Neurosci* 14(5 Pt 2):3208–24
- Sherrington CS (1910) Flexion-reflex of the limb, crossed extension-reflex, and reflex stepping and standing. *Journal of Physiology-London* 40:28–121
- Slotine JJ, Lohmiller W (2001) Modularity, evolution, and the binding problem: a view from stability theory. *Neural Netw* 14(2):137–45
- Soffe S, Roberts A (1982) Tonic and phasic synaptic input to spinal cord motoneurons during fictive locomotion in frog embryos. *Journal of Neurophysiology* 48(6):1279–1288
- St-Onge N, Qi H, Feldman A (1993) The patterns of control signals underlying elbow joint movements in humans. *Neurosci Lett* 164:171–174
- Staud G, Dengler R, Wolf W (2002) The discontinuous nature of motor execution ii. merging discrete and rhythmic movements in a single-joint system - the phase entrainment effect. *Biological Cybernetics* 86(6):427–443
- Stein P, Smith J (2001) Neural and biomechanical control strategies for different forms of vertebrates hindlimb motor tasks. In: Stein P, Stuart D, Selverston A (eds) *Neurons, Networks and Motor Behavior*, Cambridge, MA: MIT press
- Stein P, Grillner S, Selverston A, Stuart DE (1997) *Neurons, Networks and Motor Behavior*. MIT press
- Stein RB (2008) The plasticity of the adult spinal cord continues to surprise. *The Journal of Physiology* 586(12):2823–2823
- Sternad D (2007) Rhythmic and discrete movements - behavioral, modeling and imaging results. In: Fuchs A, Jirsa V (eds) *Coordination Dynamics*, Springer
- Sternad D, Dean W, Schaal S (2000) Interaction of rhythmic and discrete pattern generators in single joint movements. *Hum Mov Science* 19:627–665
- Strick P (2002) Stimulating research on motor cortex. *Nature Neuroscience* 5(8):714 – 715
- Strogatz SH (2001) *Nonlinear Dynamics and Chaos: With Applications to Physics, Biology, Chemistry and Engineering*. Perseus Books Group
- Stulp F, Oztop E, Pastor P, Beetz M, Schaal S (2009) Compact models of motor primitive variations for predictable reaching and obstacle avoidance. In: *IEEE–RAS International Conference on Humanoid Robots (HUMANOIDS 2009)*

- Suzuki R, Katsuno I, Matano K (1971) Dynamics of 'neuron ring'. *Biological Cybernetics* 8(1):39–45
- Taga G (1994) Emergence of bipedal locomotion through entrainment among the neuro-musculo-skeletal system and the environment. *Physica D: Nonlinear Phenomena* 75(1-3):190–208
- Tang W, Zhang W, Huang C, Young M, Hwang I (2008) Postural tremor and control of the upper limb in air pistol shooters. *Journal of Sports Sciences* 26(14):1579–1587, PMID: 18979336
- Thelen E, Cooke DW (1987) Relationship between newborn stepping and later walking: A new interpretation. *Developmental Medicine & Child Neurology* 29(3):380–393
- Ting LH, Macpherson JM (2005) A limited set of muscle synergies for force control during a postural task. *J Neurophysiol* 93(1):609–13
- Todorov E, Li W, Pan X (2005) From task parameters to motor synergies: A hierarchical framework for approximately-optimal control of redundant manipulators. *Journal of robotic systems* 22(11):691–710
- Tran MT (2009) Approche neuro-robotique pour la commande de gestes d'atteinte sur les robots humanoïdes. PhD thesis, Université Paul Sabatier, Toulouse
- Tresch M, Saltiel P, Bizzi E (1999) The construction of movement by the spinal cord. *Nature Neuroscience* 2:162–167
- Treuille A, Lee Y, Popović Z (2007) Near-optimal character animation with continuous control. In: *ACM SIGGRAPH 2007 papers*, ACM, San Diego, California, p 7
- Tsagarakis N, Metta G, Sandini G, Vernon D, Beira R, Becchi F, Righetti L, Santos-Victor J, Ijspeert A, Carrozza M, Caldwell D (2007) iCub - The Design and Realization of an Open Humanoid Platform for Cognitive and Neuroscience Research. *Journal of Advanced Robotics, Special Issue on Robotic platforms for Research in Neuroscience* 21(10):1151–1175
- van Mourik AM, Beek PJ (2004) Discrete and cyclical movements: unified dynamics or separate control? *Acta Psychol (Amst)* 117(2):121–38
- Vukobratovic M, Juricic D (1969) Contribution to the synthesis of biped gait. *Biomedical Engineering, IEEE Transactions on BME*-16(1):1–6
- Wächter A, Biegler LT (2006) On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming* 106:25–57

Bibliography

- Wagner D, Schmalstieg D (2007) Artoolkitplus for pose tracking on mobile devices. In: Proceedings of 12th Computer Vision Winter Workshop (CVWW'07)
- Wierzbicka M, Staude G, Wolf W, Dengler R (1993) Relationship between tremor and the onset of rapid voluntary contraction in parkinsons disease. *Jour of Neurology, Neurosurgery, and Psychiatry* 56:782–787
- Williamson M (1999) Robot arm control exploiting natural dynamics. PhD thesis, MIT Department of Electrical Engineering and Computer Science
- Wolpaw JR, Tennissen AM (2001) Activity-dependent spinal cord plasticity in health and disease. *Annual Review of Neuroscience* 24(1):807–843
- Wolpert DM, Kawato M (1998) Multiple paired forward and inverse models for motor control. *Neural Networks* 11(7-8):1317–1329
- Won J, Hogan N (1995) Stability properties of human reaching movements. *Experimental Brain Research* 107(1):125–136
- Zico Kolter J, Ng AY (2009) Task-space trajectories via cubic spline optimization. In: Proceedings of the 2009 IEEE international conference on Robotics and Automation, IEEE Press, Kobe, Japan, pp 2364–2371
- Zucker M, Bagnell JAD, Atkeson C, Kuffner J (2010) An optimization approach to rough terrain locomotion. In: IEEE Conference on Robotics and Automation

CURRICULUM VITAE

Personal Data

Name	Sarah Degallier
Birth	Le Sentier (VD), October 15th, 1979
Nationality	Swiss
Address	Biorobotics Laboratory EPFL -STI -IBI- BIOROB Station 14 CH-1015 Lausanne
Phone	(+41) 21 693 26 76
Fax	(+41) 21 693 37 05
Email	sarah.degallier@epfl.ch

Education

2006-2010	PhD candidate at the Biorobotics Laboratory, EPFL
2000-2005	Master in Mathematics, EPFL Master thesis in the Chair of Applied Statistics "Evaluation of the differences between two allelic spectra"
2002-2003	Exchange year in KTH, Stockholm, Sweden
1998-2000	Demi-licence in Psychology (2-years undergraduate program) University, Lausanne, Switzerland

Publications

Journal Articles

- Degallier S, Ijspeert AJ (2010) Modeling Discrete and Rhythmic Movements through Motor Primitives: A Review, Accepted for publication in *Biological Cybernetics*.

- Degallier S, Righetti L, Gay S, Ijspeert AJ (2010) Towards simple control for complex, autonomous robotic applications: Combining discrete and rhythmic motor primitives. Submitted to *Autonomous Robots*.

Conference Papers

- Gay S, Degallier S, Pattacini U, Ijspeert AJ, Santos J (2010) Integration of vision and central pattern generator based locomotion for path planning of a nonholonomic crawling humanoid robot. In: *Proceedings of the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2010)*, Tapei
- Degallier S, Righetti L, Natale L, Nori F, Metta G, Ijspeert AJ (2008) A modular bio-inspired architecture for movement generation for the infant-like robot iCub. In: *Proceedings of the second IEEE RAS / EMBS International Conference on Biomedical Robotics and Biomechatronics, BioRob*
- Gams A, Degallier S, Ijspeert AJ., Lenarcic, L. (2008). Dynamical system for learning the waveform and frequency of periodic signals - application to drumming. In: *15th International Workshop on Robotics in Alpe-Adria-Danube Region (RAAD)*.
- Mudry P.-A., Degallier S, Billard A. (2008). On the influence of symbols and myths in the responsibility ascription problem in roboethics - A roboticist's perspective. In: *Proceedings of the 17th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN'08)*, Munich, August.
- Degallier S, Ijspeert, AJ (2008) A bio-inspired architecture for movement generation based on discrete and rhythmic movement primitives modeled by dynamical systems. In: *AMAM 2008 abstracts*.
- Degallier S, Righetti L, Ijspeert AJ (2007) Hand placement during quadruped locomotion in a humanoid robot: A dynamical system approach. In: *IEEE-RAS International Conference on Intelligent Robots and Systems (IROS07)*
- Degallier S, Santos CP, Righetti L, Ijspeert A (2006) Movement generation using dynamical systems: a humanoid robot performing a drumming task. In: *IEEE-RAS Inter. Conf. on Humanoid Robots*, pp 512-517